

Exploiting Unstable Paths in Urban-Scale Wireless Sensor Networks

Farrukh Mirza

A Dissertation submitted to the University of Dublin, Trinity College
in fulfillment of the requirements for the degree of
Doctor of Philosophy (Computer Science)

April 2016

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Farrukh Mirza

Dated: April 29, 2016

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this Dissertation upon request.

Farrukh Mirza

Dated: April 29, 2016

Dedicated to my parents, my wife and my daughter.

Acknowledgements

First and foremost, I wish to thank my supervisors, Professor Vinny Cahill and Dr. Melanie Bouroche. They played a vital role in shaping the direction of this research while providing valuable feedback over the years.

I wish to thank my parents and my wife, Javeria, who waited patiently for me to complete my research and my thesis.

Additionally, I wish to thank the faculty, staff and students of the Distributed Systems Group and the School of Computer Science and Statistics.

In no particular order, Prof. Siobhan Clarke, Dr. Stefan Weber, Eamonn Mullins, Dr. Anthony Harrington, Dr. Andronikos Nedos, Dr. Stefano Tennina, Dr. Ivana Dusparic, Dan Marinescu, Jan Curn, Arshad Beg, Irfan Nazir, Haseeb Khan, Waqas Ahmed, Dr. Atif Manzoor, Dr. Anurag Garg, Mithileash Mohan, Dr. Serena Fritsch, Declan Ballantyne, Kamran Zafar, Prof. Mario Alves, Pedro Braga, Jose Rui Simoes, to name just a few of the colleagues and friends, I express my heartiest gratitude to them for their company and support when I needed it the most.

Farrukh Mirza

University of Dublin, Trinity College

April 2016

Abstract

Wireless sensor networks (WSNs) are composed of small autonomous devices called sensor nodes, used to measure environmental phenomena. These sensor nodes are typically unattended battery-powered devices that typically have very limited storage, computation and communication capabilities. In particular, they have limited transmission range, and therefore their data need to be collected via multi-hop routes, thereby consuming energy at all intermediary nodes. Sensor nodes deployed near a gateway may experience high volumes of traffic and congestion, hence becoming a bottleneck in the network, and therefore may consume much more energy, resulting in network partitions.

In urban environments, mobile devices such as smart phones, tablets and laptops are present. These devices generally have much better resources than sensor nodes in terms of storage, processing and communication capabilities. In particular, they may be able to establish long-range communication with distant servers via the Internet. While these devices also have a limited energy budget, they can typically be recharged easily.

The presence of such mobile devices might be exploited to (i) reduce energy consumption at resource-constrained sensor nodes, (ii) bypass congested areas of the sensor network, (iii) re-establish connectivity in the case of network partition and (iv) improve end-to-end delays.

The majority of the existing work that considers the presence of mobile devices in WSNs, however, does not exploit these devices to reduce traffic in the WSN. Furthermore, existing protocols that utilize mobile devices in WSNs do not consider all the

aforementioned objectives or address how to conciliate them. In addition, in the MANET (mobile ad hoc network) community, the few protocols that redirect traffic towards more capable devices are not suitable for resource-constrained WSNs, due to their reliance on route discovery and repair protocols, and poor or non-existent congestion controls.

Using mobile devices is difficult, because they may only appear for a short time, due to mobility and user control. For this reason, the paths established by such devices are inherently unstable, rendering route discovery a redundant task. Many existing routing algorithms do not exploit routes through such devices because of their unstable nature. Also, as these devices may not always be the closest ones to the destination, pure geographic routing algorithms typically do not take advantage of them either.

This thesis explores the possibility of exploiting such unstable paths via more capable mobile devices to achieve the aforementioned objectives. More precisely, the thesis addresses the problem of finding the best next hop in a WSN in order to outsource the communication load to mobile devices while systematically balancing the achievement of congestion reduction at critical WSN nodes, energy conservation at resource-constrained nodes, reduction in end-to-end delays and fault tolerance with only acceptable degradation in the data delivery ratio due to mobility and disappearance. In particular, this research investigates how protocols and mechanisms can be designed in a way to keep processing intensive tasks on mobile devices, thereby making such protocols practical for resource-constrained sensor nodes.

This thesis empirically shows that, in WSNs with mobile devices, just by placing the baseline discovery mechanism at more capable devices, the energy consumption of the sensor nodes reduces significantly, and by utilizing location awareness, the discovery time of the sensor nodes is improved. The empirical evidence also shows that by utilizing mobile devices appropriately, the communication load and congestion at resource-constrained critical sensor nodes is reduced, as well as the energy consumption and end-to-end delays. The results also show that in the case of network partitions, mobile devices provide alternative paths and increase the data delivery ratio.

Contents

Acknowledgements	v
Abstract	v
List of Tables	xiii
List of Figures	xv
Chapter 1 Introduction	1
1.1 Background	3
1.2 Motivation	6
1.3 Application Scenario	8
1.4 Challenges	11
1.5 Goals and Requirements	12
1.6 Contributions	13
1.7 Scope	15
1.7.1 Assumptions	15
1.8 Evaluation	18
1.9 Road Map	19
1.10 Summary	20

Chapter 2	Related Work	21
2.1	Mobile Device Discovery Mechanisms in WSNs	24
2.1.1	Baseline mobile device discovery mechanisms	24
2.1.2	Using dual beacons and radio channels for discovering mobile devices	26
2.1.3	Discovery performed by mobile devices	27
2.1.4	Learning based discovery protocols	28
2.1.5	Radio triggered wake up	29
2.1.6	Discussion	29
2.2	Routing Protocols in Mobile Ad hoc and Wireless Sensor Networks	31
2.2.1	Traditional Wireless Ad Hoc Network Protocols	32
2.2.2	Geographic Forwarding	36
2.2.3	Routing with Link Estimation	41
2.2.4	WSNs with Mobile Devices	44
2.2.5	Hierarchical Protocols	64
2.2.6	Delay-tolerant Protocols	70
2.2.7	Device Aware Protocols	74
2.2.8	Other	82
2.2.9	Summary	88
Chapter 3	Design	91
3.1	Design Overview	92
3.1.1	Decision Making	93
3.1.2	Performance Indicators	95
3.1.3	Neighbour discovery	98
3.1.4	Risk Feedback	99
3.2	Wireless Sensor Network Structure and Architecture	99
3.2.1	Wireless sensor network formation overview	101
3.2.2	Sensor node joining wireless sensor network	102

3.2.3	Adding a node as a child	104
3.3	Discovery Mechanism	105
3.3.1	Device discovery in IEEE-802.15.4 (2006) (Beacon-enabled mode) .	106
3.3.2	Stationary SNs periodically re-scan	107
3.3.3	Mobile devices periodically re-scan	111
3.3.4	Inverse Location-Aware Discovery	113
3.4	The STERIOD Routing Protocol	113
3.4.1	Performance Indicators	114
3.4.2	Credential and Risk Feedback	117
3.4.3	Decision Making	120
3.4.4	Decision Making at Sensor Nodes	120
3.4.5	Decision Making at Mobile Device	124
3.5	Summary	135
Chapter 4	Evaluation	139
4.1	Discovery Mechanisms Evaluation	140
4.1.1	Experimental design and setup	141
4.1.2	Results and Analysis	144
4.2	Simulation	154
4.2.1	Discovery manager	156
4.2.2	Network deployment scenarios	158
4.2.3	Simulating Packet Loss	160
4.2.4	Simulating Mobility	160
4.2.5	Simulating Energy Consumption	163
4.2.6	Limitations of the simulator	166
4.3	STERIOD Routing Protocol Evaluation	167
4.3.1	Objectives of the Evaluation	168
4.3.2	Wireless Sensor Network Configurations	169

4.3.3	Data Loss Due to Mobility And Its Remedy	174
4.3.4	Effect of the Selection Mechanism	179
4.3.5	Effect of Mobile Devices on the End-to-End Latency	184
4.3.6	Effect of Mobile Devices on the Communication Load	187
4.3.7	Effect of Mobile Devices on Fault Tolerance	191
4.3.8	Effect of Mobile Devices on Energy Consumption	195
4.3.9	Energy Consumption of Mobile Devices	197
4.3.10	Effect of the Speed of Mobile Devices	198
4.3.11	Comparative Study	202
4.3.12	Discussion	207
4.4	Summary	208
4.4.1	Discovery Mechanism	208
4.4.2	Routing Protocol	209
Chapter 5 Conclusions and Future Work		212
5.1	Achievements	213
5.2	Critical Analysis	215
5.3	Contributions	217
5.4	Future Work	218
5.5	Summary	220
Appendices		221
Appendix A Design Techniques		222
A.1	Avoiding beacon collisions	222
A.2	Finding possible children of a given node	225
Appendix B Basic Communication Components		227
B.1	Communication between more capable devices	227
B.2	Communication between sensor nodes	228

B.2.1	Upward communication	229
B.2.2	Downward communication	229
B.3	Communication between more capable devices and sensor nodes	231
B.4	Energy consumption on mobile devices with respect to different radio technologies - A personal experience	232
Appendix C Algorithms		233
C.1	Sensor node selection at mobile device	233
Appendix D Simulation		237
D.1	Simulated node hierarchy	237
D.2	Simulating mobility and location interpolation	238

List of Tables

2.1	Comparison of traditional routing protocols with envisaged goals and requirements.	35
2.2	Comparison of geographic routing protocols with envisaged goals and requirements.	40
2.3	Comparison of protocols that use link quality estimation.	43
2.4	Comparison of routing protocols with mobile devices within the context of WSNs, with envisaged goals and requirements.	63
2.5	Comparison of hierarchical routing protocols with the envisaged goals and requirements.	69
2.6	Comparison of delay-tolerant routing protocols with the envisaged goals and requirements.	73
2.7	Comparison of device-aware routing protocols with the envisaged goals and requirements.	83
2.8	Comparison of ExOR routing protocol with the envisaged goals and requirements.	85
3.1	Prioritized objectives to be fulfilled by mobile device along with the action to be taken to achieve each.	128
3.2	Sensor node selection strategies for prioritized objectives.	134

4.1	Common parameters for basic discovery mechanisms, for trajectory in Figure 4.1.	143
4.2	Common parameters for all discovery mechanisms, for trajectory in Figure 4.2.	144
4.3	Observed properties from experiments, for the trajectory in Figure 4.1. . .	147
4.4	Observed properties from experiments, for trajectory in Figure 4.2, BO=8.	148
4.5	Packet sizes for different packet types used in the simulation.	166

List of Figures

1.1	Exploiting a path via a mobile device.	7
2.1	PGR example, illustrated from (Roosta and Menzo, 2004)	38
2.2	TTDD's logical grid showing a source 'B' and a sink 'S' (Ye et al., 2002) .	46
2.3	SEAD's steiner tree to service multiple mobile sinks (Kim et al., 2003) . .	49
2.4	Construction of path from access node to the mobile sink (Kim et al., 2003)	49
2.5	Construction of path from source node to the mobile sink. The path is shortened, once the old path becomes inefficient (Kim et al., 2010).	54
2.6	Difference between mobility and connectivity graphs (Kusy et al., 2009). (a) Additional lines represent represent mobility between opposite sections of the building, representing non-local movement. (b) Direct connectivity is not present between the opposite sections of the building.	58
2.7	Example of hierarchical network architecture (Villasenor-Gonzalez et al., 2005).	65
2.8	Hierarchical Cluster Based Routing (Xia et al., 2009).	67
3.1	Solution components.	94
3.2	Cluster Tree Network Topology.	100
3.3	Sensor Node's Operations at Boot Time.	101
3.4	Sensor Node Joining WSN.	103
3.5	Adding a node as a child.	104

3.6	Representation of achieving $T_{D_{\min}}$	110
3.7	Representation of obtaining $T_{D_{\max}}$	111
3.8	Cluster tree network topology modified due to presence of mobile device.	118
3.9	Sensor node selection process at the mobile device.	132
4.1	Trajectory of the MDC in a single iteration.	141
4.2	Trajectory of the MDC, arriving 7 times in range in a single iteration.	143
4.3	Dotplot of T_D for the existing discovery mechanisms, for the trajectory in Figure 4.1.	145
4.4	Time series plot of T_D from the existing discovery mechanisms, for the trajectory in Figure 4.1.	146
4.5	Dotplot of T_D for the basic discovery mechanisms, following the trajectory in Figure 4.2.	147
4.6	Comparative dotplot of T_D for all discovery mechanisms, following trajectory in Figure 4.2 and BackOff in Table 4.2.	149
4.7	Energy consumption at SN vs. time for different SO, BackOff and discovery mechanisms, for the trajectory in Figure 4.2.	151
4.8	Energy consumption at SN vs. time for different BackOff and discovery mechanisms, for the trajectory in Figure 4.2.	152
4.9	Overview of the components developed for implementing Inverse Location-Aware Discovery mechanism and STERIOD protocol in simulated environment.	155
4.10	Variant of the strategy pattern, implemented to perform sensor node selection by the mobile device using different combinations of pre selection filter, selection strategy, and post processor, in order to achieve prioritized objectives.	157
4.11	Location generators emulating mobility on mobile devices.	161

4.12 Configuration 1: Network deployment with 28 sensor nodes and two children per node. (Two Children Per Node - C1 - 28SN)	169
4.13 Configuration 2: Single branch network deployment with 12 sensor nodes. (Single Branch - C2 - 12SN)	170
4.14 Configuration 3: Unbalanced network deployment with 14 sensor nodes. (Unbalanced Network - C3 - 14SN)	170
4.15 Configuration 4: Planned hierarchical network deployment with 61 sensor nodes. (Hierarchical Designed Network - C4 - 61SN)	171
4.16 Configuration 5: Half connected network deployment with 15 sensor nodes. (Half Connected Network - C5 - 15SN)	172
4.17 Configuration 6: Disconnected network deployment with 14 sensor nodes. (Disconnected Network - C6 - 14SN)	172
4.18 Configuration 7: Semi automatic circular network deployment with 137 sensor nodes. (Semi Automatic Circular - C7 - 137SN)	173
4.19 Delivery ratio, with and without transmission retries, using network configuration C1.	175
4.20 Delivery ratio, with and without transmission retries, using network configuration C2.	176
4.21 Transmission required by sensors per generated packet, with and without transmission retries, using network configuration C2.	177
4.22 Transmission overhead ratio per generated packet, with and without transmission retries, using network configuration C2.	177
4.23 Transmission required by sensors per generated packet, with and without transmission retries, using network configuration C1.	178
4.24 Transmission overhead ratio per generated packet, with and without transmission retries, using network configuration C1.	179

4.25	Results using different selection mechanisms showing better performance with the traffic reduction selection mechanism, using the network configuration C7.	181
4.26	Results using different selection mechanisms showing better average hop counts with the traffic reduction mechanism but better delivery ratios with the credential improvement selection mechanism, using the network configuration C7.	182
4.27	Energy consumption using different selection mechanisms, using the network configuration C4.	182
4.28	Transmissions required by the sensor and transmission overhead ratio per generated packet using different selection mechanisms, using the network configuration C2.	183
4.29	Average number of hop counts traversed by generated packets, using connected network configurations C1, C2, C3, C4 and C7.	185
4.30	Average number of hop counts traversed by generated packets, using network configurations C5 and C6.	186
4.31	Transmission ratio by sensors, using the network configurations C1, C2, C3, C4 and C7.	188
4.32	Transmission ratio by cluster heads, using network configurations C1, C2, C3, C4 and C7.	189
4.33	Transmissions required by sensors per generated packet, using network configurations C1, C2, C3, C4 and 7.	190
4.34	Transmissions overhead ratio by sensors per generated packet, using network configurations C1, C2, C3, C4 and C7.	191
4.35	Delivery ratio, using the network configurations C5 and C6.	192
4.36	Transmission ratio by cluster heads, using the network configurations 5 and 6.	193

4.37	Transmissions required by sensors per generated packet, using the network configurations 5 and 6.	194
4.38	Energy consumption at sensor nodes in milliamperes, using network configurations C1, C2, C3, C4 and 7.	195
4.39	Energy consumption at cluster heads in milliamperes, using network configurations C1, C2, C3, C4 and C7.	196
4.40	Delivery ratio for different maximum speed and mobile devices, using the network configuration C7.	199
4.41	Transmission overhead ratio per generated packet for different maximum speed and mobile devices, using the network configuration C7.	199
4.42	Delivery ratio and transmission overhead ratio per generated packet for different maximum speed with 10 mobile devices, using the network configurations C1, C2, C3, C4 and C7.	200
4.43	Delivery ratio and transmission overhead ratio per generated packet for different maximum speed with 50 mobile devices, using the network configurations C1, C2, C3, C4 and C7.	201
4.44	Energy consumption with increasing number of P-nodes (Liu et al., 2011).	202
4.45	Delivery ratio with increasing number of P-nodes (Liu et al., 2011).	203
4.46	Average end-to-end delay with increasing number of P-nodes (Liu et al., 2011).	203
4.47	The delivery ratio with increasing number of mobile nodes, using the network configurations C1, C2, C3, C4 and C7.	204
4.48	Average end-to-end delay with increasing maximum speed of P-nodes (Liu et al., 2011).	206
4.49	Delivery ratio with increasing maximum speed of P-nodes (Liu et al., 2011).	206
A.1	Beacon Separation in Time Domain to Avoid Collisions.	223
A.2	Spatial Limitation on Beaconing Devices.	225

B.1	Data transfer to a parent node (Coordinator) from a child node (Network Device), IEEE802.15.4 (2006).	228
B.2	Data transfer to a child node (Network Device) from the parent node (Coordinator), IEEE802.15.4 (2006).	229
B.3	Data transfer to a child node (Network Device) from the parent node (Coordinator), as a broadcast packet with addressing fields in the network layer header.	230
B.4	Fixed gateway (host) connected with a sensor node over USB communicates with other sensor nodes over wireless link.	231
D.1	Representing different types of nodes in simulated environment.	238

Chapter 1

Introduction

Wireless sensor networks (WSNs) are composed of resource-constrained autonomous sensor nodes, used to measure environmental phenomena. They are typically unattended battery-powered devices that have very limited storage, computation and communication capabilities. Generally their transmission range and the number of bits per packet are also limited to conserve energy, and therefore their data need to be collected via multi-hop routes, thereby consuming energy at all intermediary nodes.

Frequently hierarchical sensor network deployments are preferred to allow the network to scale, such that the sensor data is delivered to a central device (sink), which has significantly higher resources than the sensor nodes. However, in a hierarchical network structure, sensor nodes deployed near the sink may experience congestion due to high volumes of incoming traffic. In addition, the energy reserves on such sensor nodes can deplete more quickly than at other sensor nodes in the network due to frequent data forwarding.

Communication is typically considered to be the core reason behind energy consumption on sensor nodes as the energy consumed during processing tasks is comparatively negligible. Reduction in communication based on complex data aggregation, compression, or stochastic prediction methods (Tennina et al., 2011a; Chu et al., 2006; Mohan

et al., 2012; Marcelloni and Vecchio, 2008; Kimura and Latifi, 2005; Mohamed et al., 2013) may not always be suitable on the resource-constrained sensor nodes.

In modern urban environments, however, more capable mobile devices are ubiquitously present. These devices usually have much better resources than sensor nodes. Therefore, this thesis explores the possibility of exploiting such mobile devices to reduce communication in a WSN, thereby reducing energy expenditure. The more capable mobile devices, however, are individually owned and not under the control of the deployed sensor network. Thus, these devices can appear and disappear at any time, either because they move fast (e.g., in vehicles) or because the user (owner of the device) changes the state of the radio responsible for communicating with the sensor network.

Thus, in this thesis, a new approach is developed in which unstable links through the more capable mobile devices are exploited and the characteristics of each of the available devices are used to differentiate between their capabilities, to reduce network traffic within a WSN. Hence, this thesis addresses the problem of offloading communication load from unattended sensor nodes to more capable mobile devices, in order to conserve energy, reduce congestion, provide fault tolerance and improve latency.

In particular, the Inverse Location-Aware Discovery (ILAD) mechanism presented in this thesis, demonstrates how mobile devices can be discovered quickly without compromising the energy efficiency of the sensor nodes. In addition, the Smart Traffic Energy and Resource aware Offloading using Inverse Discovery (STEROID) protocol shows how communication tasks can be offloaded in urban-scale WSNs while achieving the aforementioned objectives.

This thesis studies the relationships and tradeoffs between different parameters in order to achieve the aforementioned goals and investigates how the risk of data loss can be limited while using mobile devices in the STEROID protocol. As a result, the STEROID protocol outsources the communication to extra-WSN devices, improving end-to-end latency and energy conservation at the sensor nodes, in addition to reducing congestion at the critical nodes while also providing fault tolerance in the case of network

partitions.

The rest of this chapter is organized as follows. Section 1.1 provides background on WSNs and participatory sensing and briefly discusses problems associated with each. Section 1.2 motivates the research in this thesis by discussing the problems associated with typical WSNs. Section 1.3 discusses a hypothetical application scenario for the work presented in this thesis. Section 1.4 discusses the challenges associated with this research and formulates the problem. Section 1.5 maps out the requirements associated with the envisaged solution. Section 1.6 discusses the contributions of this thesis. Section 1.7 formulates the scope of this thesis. Section 1.8 discusses the evaluation plan to evaluate the contributed protocol. Section 1.9 briefly discusses the roadmap of this thesis. Finally Section 1.10 summarizes this chapter.

1.1 Background

Verdone et al. (2010) define a Wireless Sensor and Actuator Network (WSAN) or WSN as a network of small (in size) and simple devices (sensor nodes) that can sense the environment and communicate the gathered information through wireless links. The data is forwarded to a sink via (possibly) multi-hop routes. The sink is typically defined as a single node within the WSN, that collects all the sensor data from the sensor nodes that are directly or indirectly connected to it. The sink can use the data either locally or further forward it to external networks (e.g., the Internet) via a gateway. The gateway is defined as a device that bridges two (or more) distinct types of networks together, e.g., a gateway can become a bridge between a WSN and the Internet. In this thesis, we use the term “*Fixed Gateway*” to represent a stationary device that is a combination of a typical WSN sink and a gateway connected to an external network (e.g., the Internet). Assuming the presence of a fixed gateway allows the WSN to be connected to the external network at all times (Tennina et al., 2011a).

In general, sensor nodes are low-power autonomous devices with very limited storage,

computation and communication capabilities. Apart from sensor data delivery, energy conservation is usually considered to be the most important requirement for the WSN, because the sensor nodes are battery powered and unattended devices. Recharging the batteries of the sensor nodes is typically impractical as the sensor nodes are deployed in hard-to-access areas and physical access can incur high maintenance cost (Dohler et al., 2007; Anastasi et al., 2009). Therefore, the protocols for WSN are typically designed to ensure energy conservation.

In addition, WSN nodes also have limited transmission range and can typically only send packets of small sizes. For example, the IEEE802.15.4 (2006) standard defines a maximum packet size of 127 bytes that the physical layer shall be able to receive. In order to conserve energy, the transmission range can be further limited by reducing the transmission power. For this reason, multiple hops are used to route data towards the sink.

Typically, data communication consumes the most energy reserves (Anastasi et al., 2009; Pottie and Kaiser, 2000; Raghunathan et al., 2002; Chipcon, 2006). For this reason, the data communication is minimized at all levels of the software stack and algorithms are optimized for energy efficiency. The typical life cycle of the sensor node's radio is designed to alternate between active and sleep modes, such that the sensor node switches off its radio to conserve energy during the sleep mode and listens for incoming traffic during the active mode. The duty cycle of the on-board radio is configured in a way that keeps the radio in sleep mode for the majority of the time. Data communication is minimized using different techniques, at different levels of the software stack. For example, IEEE802.15.4 (2006) has listening duty cycles in beacon-enabled mode to limit the portion of time a node spends listening to the radio channel, routing protocols are designed to be energy-aware (Yu et al., 2001; Roosta and Menzo, 2004; Akkaya and Younis, 2004; Avudainayagam et al., 2003), and middleware systems are designed to reduce packet transmissions either by sensor data aggregation (Tennina et al., 2011a), utilizing stochastic data correlation models (Chu et al., 2006; Mohan et al., 2012), or

by compressing the sensor data (Marcelloni and Vecchio, 2008; Kimura and Latifi, 2005; Mohamed et al., 2013). In multi-hop WSNs, the sensor nodes not only transmit their own data but also forward the data on behalf of other sensor nodes. Forwarding data on behalf of other sensor nodes generally results in higher energy consumption at the intermediary sensor nodes, as it generates extra transmissions.

The research community is advocating participatory and opportunistic mobile sensing, where sensors embedded in mobile devices of individual users could be used to gather environmental information (Burke et al., 2006; Mun et al., 2009; Lane et al., 2008). This approach, however, is only suitable for cheap common sensors, as otherwise device costs may become prohibitive. Data availability also highly depends on the availability of participating devices in such cases, which means that if a device is not present in the target area, then sensor readings cannot be gathered. For these reasons, participatory sensing may be utilised for gathering generalised information, e.g., temperature and noise, for which sensors can easily be embedded within consumer devices. This information could then be used in non-critical applications. However, this approach is not suitable for applications that require specialised sensors, e.g., CO_2 , NO_2 and O_3 , due to the price of such sensors, or for mission critical applications, where data could be required at any time regardless of the presence of participating devices in the target area. WSNs are expected to be useful in the applications where such specialised sensor readings are required.

Overall, WSNs are suitable for sensing (specialized) environmental phenomenon without depending on the availability of participating and opportunistic devices. Whereas, simultaneously reducing communication activity on resource-constrained sensor nodes is desirable to reduce energy consumption.

1.2 Motivation

In large-scale WSNs, a number of challenges are faced, such as energy conservation, handling traffic congestion, fault tolerance, maximizing data delivery ratio, timeliness and scalability, among others (Anastasi et al., 2009; Wan et al., 2003; Tennina et al., 2011a). Most of these challenges are due to the limited capabilities of the devices used to form such networks. All of these challenges ultimately affect the timely data delivery to the final destination. As the number of hops increase in the sensor network, the end-to-end latency increases. In addition, in typical deployments, the data is directed towards the fixed gateway (GW) (Somasundara et al., 2006; Tennina et al., 2011a). Therefore, packets from many sources converge towards nodes that are closer to the fixed gateway, resulting in traffic congestion at these nodes (Somasundara et al., 2006; Wan et al., 2003; He et al., 2003). These nodes, thus form critical resources in the network, and may become bottlenecks. Due to higher volumes of data forwarding, these critical nodes may drain their batteries comparatively quickly compared to the rest of the nodes. Once these nodes become non-operational, severe network partitioning can be witnessed, such that data from source nodes may not reach fixed gateway and beyond.

In urban environments, mobile devices such as smart phones, tablet PCs and laptops are abundantly present. These devices generally have much better resources than sensor nodes in terms of storage, processing and communicating capabilities. These devices may also be able to establish long-range communication with distant servers via the Internet. Since these devices are owned by individuals, even though they have limited energy reserves, they can be recharged easily. Due to their higher capabilities, these devices can be used to collect and process data from sensor networks. In addition, these devices can be used to opportunistically forward data on behalf of sensor nodes in an opportunistic and dynamic manner (Figure 1.1).

To alleviate the challenges associated with communication in WSNs, this thesis considers how mobile devices can be opportunistically exploited to offload packet forwarding

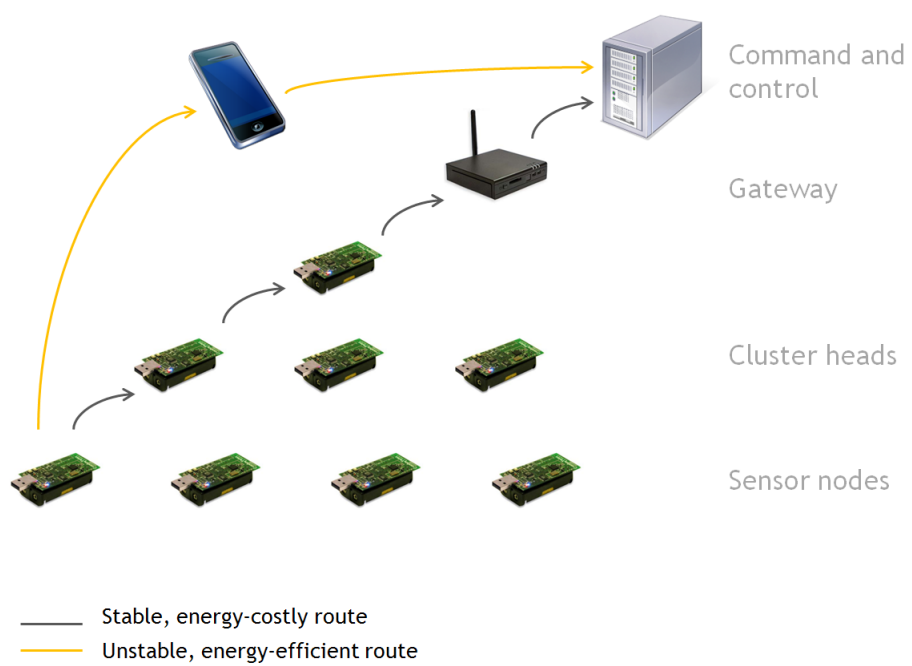


Fig. 1.1: Exploiting a path via a mobile device.

and information dissemination tasks from resource-constrained sensor nodes. By taking advantage of the paths created by participating mobile devices and using them as mobile gateways, the data forwarding and routing load can be moved away from the deployed WSN to more capable mobile devices. Shifting the communication, processing and data forwarding tasks from resource-constrained sensor nodes to the aforementioned mobile devices, the burden on the sensor nodes can be reduced. This technique can essentially reduce the energy consumption on the sensor nodes, because now the sensor nodes do not have to frequently forward data on behalf of other sensor nodes, possibly resulting in longer network lifetime. Longer network lifetime generally translates into better return on investment (ROI), reducing the operational costs of the WSN in terms of changing batteries or deploying new sensor nodes when old ones deplete. Also, congested areas of the sensor network can be avoided by forwarding the data through these mobile devices. In case of network partitions, mobile devices can be used to re-establish connectivity from disjoint portions of the sensor network. In addition, significant improvements are expected to be achieved in end-to-end delays due to higher transmission rates and direct connectivity to the destination via the Internet, if long range radio is available.

This hybrid strategy can potentially leverage the best of both paradigms, as absolute reliance on deployed infrastructure to deliver information is reduced, while gathering environmental information from sensor network deployment without relying on the availability of participatory sensors in the target region.

1.3 Application Scenario

The work presented in this thesis can be applied to an application scenario where a company (such as a telecommunication network operator) decides to deploy geographically dispersed wireless sensors throughout an urban center. The sensor data could be collected, for example, to understand the impact of marketing campaigns on consumer behaviour, e.g., how certain marketing material such as billboards or posters, offering

higher value for the customers attract people towards the company's retail outlets, and if this results in actual sales. The company can also aim to expose the sensor data to other companies or city council who wish to take advantage of such data, for example, to build detailed real-time noise and pollution map for city planning purposes. In such scenario, multiple companies along with city councils can cooperate to invest in such an implementation of the sensor network.

In order to reduce the cost, the company chooses low cost wireless sensors to monitor a number of environmental phenomena such as air, light and noise pollution, temperature and humidity, proximity (using infrared proximity sensors) of pedestrians and vehicles with the sensors deployed throughout the city. In order to reduce the cost associated with connecting fixed power lines throughout the network, the company is likely to opt for battery-operated sensors.

In order to achieve higher sensor data fidelity, the company opts to deploy sensors quite close to each other, say approximately every 5-10 meters. The company decides to organize the sensor network in cluster-tree network topology in order to allow the network to scale easily (Tennina et al., 2011a,b). In such a scenario, the data acquired by the sensors is delivered to a centralized location to understand consumer behaviour by combining sales data and pedestrians' mobility patterns with marketing campaigns, weather and pollution levels (out of the scope of this research). The environmental sensor data is also exposed to the wider research community and the industry at large for a nominal fee by exposing a set of easy to use APIs (application program interface).

However, the company realizes that a number of problems emerge in wireless sensor network due to their less capable nature as discussed in the previous sections. In order to improve ROI, the company decides to crowd source sensor data communication and delivery by utilizing their telecommunication network consumer base. The communication is not completely relied upon crowd sourcing because the company is interested in acquiring the sensor data even if no data forwarding participants are available, i.e., the data is forwarded via the normally connected sensor network, for example, in the case of

unavailability of participants or if the communication via participatory mobile devices has higher risk of data loss. In order to increase the number of participants who allow their smart mobile devices to become the data carriers, the telecommunication company offers tangible real-world rewards in exchange for consumer's participation, e.g., free talk time, extra data bundles, loyalty points (e.g., Tesco club card points when data is forwarded via Tesco mobile subscribers, hypothetically) or free meal at a partner restaurant. Since the more capable mobile devices are owned by independent participants, they can recharge their devices as required, e.g., when they connect their device to their car or they can re-charge their device when they reach their destination. In addition, the participants can allow the software published by the said company to use their device to forward sensor data when they have enough battery remaining on their devices. For these reasons, this thesis does not consider energy consumption optimization at the mobile devices to be one of its crucial requirements.

In such a scenario, the location of the sensor nodes, along with their transmission range, will be known a priori to the company deploying sensors throughout the city, as the sensors will be considered as assets of the company, representing significant investment. The company can share the location of the sensors with the smart mobile devices that offer their services to forward data on behalf of the sensor network using the ILAD mechanism and the STERIOD protocol.

This thesis assumes compatible radios on both more capable mobile devices and less capable sensor nodes where the transmission range of the sensor nodes is limited in accordance with desired operational life in terms of battery consumption. Because, the less capable sensor nodes will not be allowed to increase their transmission power to respond to mobile devices by sending acknowledgements, even if the mobile devices are allowed to increase their transmission power to send packets to sensors deployed farther than their normal range, the acknowledgement packets originating from the sensors will have to be traversed using multi-hop paths. In such a scenario, the mobile device may never receive acknowledgement packets, because by the time such packets are routed to the

mobile device, it may have moved and the path to the mobile device may have changed. In essence, by allowing uni-directional links, the company will have to implement a solution that is similar to the ones discussed in Chapter 2, Section 2.2.4, just to deliver acknowledgement packets to the mobile devices. The uni-directional links also increase the chances of loss of acknowledgement packets as the probability of transmission error will increase proportionally with the increase in the number of intermediary devices while routing acknowledgement packets to the mobile device. For these reasons, there is no need to allow the mobile devices to be able to communicate with sensors that are farther away by increasing their transmission range in similar scenarios, thus reducing the implementation cost.

1.4 Challenges

The most prominent characteristic of the paths introduced by these mobile devices is that they are highly unstable. Indeed they can appear when mobile devices are present in the vicinity and disappear as soon as any of the mobile devices within that path moves outside the communication range of the other devices. The appearance and disappearance of devices could also be triggered when the user of such a mobile device turns on or off the radio responsible for communicating with the WSN. For this reason, using such mobile devices to forward data increases the risk of data loss. Therefore, a tradeoff exists between energy consumption, end-to-end delay, throughput and risk of data loss when using these mobile devices.

Thus, this thesis addresses the problem of finding the best next hop in a WSN in order to move data forwarding and processing load to more capable mobile devices, while conserving energy, reducing congestion, providing fault tolerance and improving end-to-end latency. This thesis also studies how these objectives can be achieved simultaneously while limiting the risk of data loss due to the mobility of mobile devices.

In addition, the proposed approach of outsourcing data forwarding and communica-

tion load to more capable mobile devices introduces the challenge of finding and then using these devices efficiently. This means that the mobile devices must be discovered by the stationary, less capable, sensor nodes as quickly as possible, to take advantage of the potential routes provided by the mobile devices. In addition, the discovery mechanism must also be energy efficient, especially at the sensor nodes in order to keep the sensor nodes operational for longer durations.

As the sensor nodes are very resource-constrained devices with very limited memory and processing capabilities, they cannot store information about all mobile devices that may become available in their vicinity to take intelligent decisions, as high volumes of mobile devices may be available for exploitation in urban environments with different arrival patterns. Thus, only limited amounts of information can be stored temporarily for decision making, in order to reduce memory and processing requirements.

1.5 Goals and Requirements

The main goal of this thesis is to investigate how to efficiently use mobile devices available in urban environments to outsource data communication load from resource-constrained sensor nodes in the network.

This thesis sets the following requirements while achieving the above-mentioned goal:

1. **Device Aware:** The routing and discovery protocols must be device aware such that they can differentiate among different types of devices and their capabilities.
2. **Latency Reduction:** The routing protocol must minimize end-to-end latency by reducing end-to-end packet delivery time for timely delivery of data packets.
3. **Energy Conservation:**
 - (a) The discovery protocol must be energy efficient at the sensor nodes without compromising sensor network lifetime while leveraging mobile devices.

- (b) The routing protocol must increase network lifetime by reducing energy consumption at the stationary intermediary sensor nodes, i.e., by reducing packets forwarded by these nodes, and by using energy-aware decision metrics.
- 4. **Congestion Reduction:** The routing protocol must decrease congestion at critical sensor nodes, i.e., reduce incoming traffic at sensor nodes engaged in high data forwarding to increase productivity of the critical resource-constrained sensor nodes.
- 5. **Fault Tolerance:** The routing protocol must provide fault tolerance in the case of network partitions by taking advantage of mobile devices.
- 6. **Fast Reactivity:** The discovery protocol must quickly discover mobile devices in order to allow the sensor network to leverage their presence in a timely manner.
- 7. **Communication Load Reduction:** The routing protocol must offload communication load from resource-constrained sensor nodes by redirecting traffic towards more capable mobile devices.
- 8. **Data Delivery Ratio:** The routing protocol must limit the risk of data loss while forwarding via mobile devices due to their mobility.

1.6 Contributions

The research described in this thesis contributes to the state-of-the-art in the area of routing and device discovery within the domain of WSNs deployed in urban environments.

The core contributions of this thesis are as follows:

1. Extensive analysis of the state-of-the-art mobile device discovery protocols within the context of WSNs, and closely-related routing protocols that utilize mobile devices in different capacities encompassing the multiple domains of mobile ad hoc

networks (MANETs), delay tolerant networks (DTNs) and wireless sensor networks (WSNs). The analysis of the related work shows that none of the existing protocols can be used in resource-constrained WSN environment to either discover mobile devices quickly and energy-efficiently, or offload the communication and processing load to more capable mobile devices to achieve energy conservation, fault tolerance, congestion reduction and end-to-end latency reduction simultaneously with limited risk of data loss.

2. A new location-aware device discovery mechanism (ILAD) that allows sensor nodes and mobile devices to discover each other efficiently. This discovery mechanism moves the discovery procedure to the more capable mobile devices. It achieves faster discovery by exploiting location awareness at the mobile devices and improved energy consumption at the sensor nodes when compared to the baseline discovery mechanism. An implementation of the discovery mechanism is provided on sensor nodes in TinyOS (Levis and Gay, 2009) on the TelosB (Crossbow, 2005) platform and on mobile devices in Java, to understand its characteristics in a test-bed environment, such as discovery time and energy consumption at the sensor nodes.
3. A new routing protocol (STEROID) that allows the more capable mobile devices to take over data forwarding activity from sensor nodes. This routing protocol reduces end-to-end latency by avoiding multi-hop routes via sensor nodes. It reduces congestion by reducing packet arrivals at critical nodes. In addition, it also improves energy consumption at intermediary nodes while reducing packet arrivals at these intermediary nodes, and provides fault tolerance in the case of network partitions. An implementation of the routing protocol is provided in a custom Java based simulator that is portable on multiple operating systems, to understand its impact on energy consumption, transmission overhead, delivery ratio, transmissions required to deliver packets and transmissions by intermediate sensor nodes

under different network deployment scenarios.

1.7 Scope

This thesis mainly focuses on the routing problem in WSNs to offload communication related tasks from resource-constrained sensor nodes to more capable mobile devices. In particular, this thesis focuses on WSNs deployed in urban environments within the smart cities context, where mobile devices, such as smart phones, tablet and laptop computers, are abundantly present.

The research aims at reducing the data communication within the sensor network by outsourcing the communication traffic originating from sensor nodes to extra-WSN (cooperative) devices while bypassing the intermediary sensor nodes. The discovery mechanism to discover mobile devices is within the scope of this thesis. In this work, the mobile devices are used as mobile gateways to forward packets containing sensor readings to a centralized command and control server (CC). However, the server side network structure along with the security and privacy issues related to open access networks are out of the scope of this thesis.

The mobile devices could be extended to act as sinks as well, in order to allow them to also gather sensor readings. However, mobile sinks are out of the scope of this thesis, as many existing protocols target supporting mobile sinks as their primary objective (Akkaya and Younis, 2004; Kim et al., 2003; Ye et al., 2002; Kim et al., 2010; Kusy et al., 2009).

1.7.1 Assumptions

This thesis makes the following assumptions about the environment:

1. The WSN is organized in a hierarchical cluster-tree network topology. WSNs can alternatively be deployed in a mesh network topology, however, existing work

(Tennina et al., 2011a,b) suggests that WSNs deployed in a hierarchical cluster-tree network topology scale well.

2. All the sensor data is forwarded towards a centralized command and control server (CC) that is not co-located with the sensor network. The centralized CC can physically comprise of multiple servers or virtual servers, but is assumed to be logically exposed by a single end-point (IP address or URL) and is therefore, referred to as a single unit. The location (URL end point) of this centralized CC is assumed to be accessible via the Internet. Alternatively, the sensor data can be consumed locally within the vicinity of the sensor network or can be forwarded to different locations, however, accumulating sensor data in a centralized location allows third-party software vendors to consume the sensor data by utilizing a set of well known APIs such as webservice.
3. In order to gain targeted insight, once the sensor network is formed, all sensor nodes start transmitting their sensor readings towards the fixed gateway, which in turn forwards these readings to the command and control server.
4. Any third-party consumers of the sensor data can fetch the relevant information from the command and control server connected to the Internet.
5. Each node in the sensor network knows its own location either by deployment time configuration or by using localization techniques (Caballero et al., 2008; Graefenstein et al., 2009; Priyantha et al., 2005). The sensor nodes include their own geographic location in their periodic beacon broadcast, that is used by the mobile devices to perform traffic reduction by inferring the sensor nodes that are in range of one another.
6. The transmission range of sensor nodes is known a priori as the transmission power is configured at deployment time.

7. Mobile devices can obtain their location from an external source, such as the Global Positioning System (GPS). The mobile device uses its own location to perform location-aware discovery of the sensor nodes in its vicinity in order to reduce discovery time.
8. The mobile devices can also obtain the location of the sensor network, (possibly) even the sensor nodes, from the CC to perform location-aware discovery of the sensor nodes in its vicinity.
9. All links are symmetric and bi-directional, i.e., if device 'A' can communicate with device 'B', then device 'B' can also communicate with device 'A'. This assumption allows single hop transmissions to be bi-directional, i.e., if the data packets are delivered in a single hop then their acknowledgement packets are also delivered in a single hop. If the links are allowed to be uni-directional, i.e., the more capable device can increase its transmission power to reach farther destinations, then the data transmission can happen in a single hop, whereas, the acknowledgement packets may arrive at the source via multi-hop paths. Such mechanism requires multi-hop routing of the acknowledgement packets in the reverse direction, that may increase overall routing overhead (Du et al., 2006). If there are delays in the arrival of the acknowledgement packets via multi-hop paths or such packets are lost, then the source node may send the original data packet again in order to increase delivery probability, thus unnecessarily increasing the network traffic. If request-to-send (RTS) and clear-to-send (CTS) based communication mechanisms are used at the lower layers of the traditional OSI model, then by allowing uni-directional links, by the time CTS packet is received at the source via multi-hop routes, the channel may not actually remain clear. Due to these problems faced with uni-directional links, this thesis assumes and allows communication only via bi-directional links. This assumption, however, does not mean that link layer errors cannot occur between two communicating nodes, i.e., if device 'A' sends data

packet to device 'B', then there is no guarantee that the acknowledgement packet from device 'B' to device 'A' will not be lost or the data packet itself will not be lost.

10. The energy consumption on the mobile device does not need to be optimized as a mobile device is individually owned and thus can be easily recharged.
11. The command and control server, the fixed gateways and the mobile devices support multi-threading because they are more capable devices as compared to resource-constrained sensor nodes.

1.8 Evaluation

This thesis evaluates the routing protocol and the discovery mechanism on actual sensor nodes to ensure real-world validation, and within a simulated environment to study the impact of distinct network characteristics in isolation. The ILAD mechanism is empirically evaluated in terms of energy consumption at the sensor nodes and the interval of observed discovery time. The STERIOD routing protocol is empirically evaluated in terms of reduction of the traffic load within the sensor network, energy consumption at the sensor nodes, end-to-end latency and packet delivery ratio, within the simulated environment.

This thesis studies the impact of the protocol on the sensor network under different environments. A number of different network deployments and configurations are used to understand the impact of the location of sensor nodes, and their connectivity with respect to the fixed gateway and mobile devices. As the number of combinations of experimental parameters is high, simulations are used to conduct the study in order to reduce the configuration and experimental design time, and increase the experimental flexibility. Each experimental configuration is executed multiple times to improve statistical significance of the results.

The network deployments used in the evaluation of the routing protocol generally fall under two categories:

1. **Planned:** Manually planned deployment to study different aspects of the protocol.
2. **Generated:** Algorithmically generated to avoid human intervention and achieve an easy deployment strategy.

The proposed routing protocol (STEROID) is evaluated with randomized mobility traces, whereas the proposed discovery mechanism (ILAD) is evaluated with fixed mobility traces. Two separate mobility trace generators were implemented that generate the traces based on either fixed locations that a mobile device visits or the random waypoint model with zero (0) pause time¹.

1.9 Road Map

The remainder of this thesis is organized as follows. Chapter 2 presents the state-of-the-art in mobile device discovery and routing in ad hoc and wireless sensor networks, with a particular emphasis on protocols that consider stationary sensor networks with mobile devices, and protocols that redirect traffic towards more capable devices. It demonstrates that none of the existing work is suitable to completely fulfil the requirements set out in Section 1.5. Chapter 3 discusses in detail the design of the ILAD mechanism and the design of the STEROID routing protocol. Chapter 4 empirically validates the hypotheses discussed in this chapter and evaluates the ILAD mechanism in terms of energy consumption at the sensor nodes and discovery time, and the STEROID routing protocol in terms of average number of traversed hops, delivery ratio, transmission ratio at the sensor nodes, transmissions required by sensor nodes for delivering packets, transmission overhead at the sensor nodes and energy consumption at the sensor nodes, in a

¹A number of tools were also developed during the course of this research to easily and quickly analyse the output of the experiments.

test-bed and simulated environment respectively under different sensor network deployments. Additionally, Chapter 4 also discusses the custom simulator used in the course of this research. Chapter 5 concludes this thesis and discusses the future work associated with this research.

1.10 Summary

This chapter began by introducing WSNs and the major emphasis of the research community in the area. It then motivated the research presented in this thesis by outlining the problems associated with resource-constrained sensor networks and the approach taken to overcome those issues, i.e., to outsource data communication to individually-owned mobile devices present in urban environments. This chapter also discussed a hypothetical application scenario where ILAD and STERIOD protocols can be used. This chapter then outlined the challenges associated with this approach and the goals and requirements that are to be met in order to achieve a successful implementation. The contributions of the thesis were systematically outlined later in this chapter. The scope of this thesis along with the assumptions made by the system were also defined in this chapter. The chapter concluded by outlining the approach taken for evaluation of the system and the outline of this thesis.

Chapter 2

Related Work

The problem of shifting communication load to more capable mobile devices comprises of device discovery and routing sub problems. Existing mechanisms have used a variety of techniques to resolve these two sub problems. This chapter discusses existing work in the field of device discovery and packet routing within the context of wireless sensor networks (WSNs) and develops a taxonomy of the most relevant protocols.

Discovery protocols related to discovering mobile devices in WSNs are discussed first, along with their shortcomings. The majority of the existing discovery protocols in WSNs have placed the burden of discovery on resource-constrained sensor nodes. The discovery protocols discussed in this chapter mainly differ in the way they optimize the discovery process to quickly discover the mobile devices. This chapter first discusses the base line discovery protocols that either continuously, randomly or periodically scan for mobile devices. This chapter also discusses the discovery protocols that use either two beacons or two different radios each listening on a different radio channel, to discover mobile devices more efficiently than the baseline discovery protocols. More intelligent discovery protocols use learning techniques to predict the arrival times of the mobile devices, thus improving the state-of-the-art. Even though potent, these techniques are complex to implement on sensor nodes due to the resource-constrained nature of the sensor nodes

in terms of limited energy reserves and processing and storage capabilities. The most efficient protocols discussed in this chapter use hardware oriented techniques to wakeup the sensor nodes with which communication is intended. These protocols, however, introduce extra hardware on the sensor nodes that may result in higher manufacturing cost. As the majority of the protocols assume that the resource-constrained sensor nodes will scan for beacons originating from mobile devices, this chapter also discusses an alternative protocol in which the mobile devices scan periodically for beacons originating from the sensor nodes. This alternative technique inspires the development of the inverse location-aware discovery protocol discussed in this thesis that reduces the listening tasks performed by the resource-constrained sensor nodes while improving the discovery time when compared to the baseline discovery mechanisms.

After discussing the discovery protocols, this chapter discusses the traditional and geographic routing protocols employed in MANETs. The traditional routing protocols are discussed first because these protocols provided the basis for establishing ad hoc networks without a fixed infrastructure. However, these traditional routing protocols suffer in terms of their reactivity due to mobility because they flood the network with route requests whenever a new route is required. This drawback has been largely overcome in MANETs by employing geographic routing protocols that use the location information of their destination and their neighbourhood to route packets towards the destination. However, if the destination moves away from the known location, then the geographic routing protocols also fail to deliver data. If the destination is a server located behind an IP network, then geographic routing protocols can only deliver data to a gateway from where the data is forwarded via IP network routing protocols.

Because this thesis focuses on using mobile devices with stationary WSN, routing protocols that employ mobile devices in some capacity within the context of WSNs are discussed next. This discussion shows that the majority of the existing work focuses on routing the sensor data to specific mobile devices that use the sensor data for their own consumption. In this way, the resource-constrained sensor nodes strive to form

paths towards mobile devices and track the movement of these mobile devices in order to deliver their data. However, the existing work in this area does not use the mobile devices to offload the communication tasks from the resource-constrained sensor nodes.

Hierarchical routing protocols are explored next in this chapter because the WSN assumed in this thesis is organized in hierarchical network topology. Even though these protocols employ mobile devices and form the network as a hierarchy of heterogeneous devices, they mostly focus on reducing the information maintained in the routing tables and reducing the impact of route requests in the network by avoiding route request flooding throughout the network. The discussion on hierarchical routing protocols show that these protocols are not suitable in the scenario targeted in this research either where the communication load is to be offloaded to more capable mobile devices.

Another interesting category of routing protocols deals with disjoint and partitioned networks. Because the delay tolerant routing protocols focus on sharing information when devices arrive in the proximity of each other, they also show similarity with the scenario presented in this research. Thus delay tolerant routing protocols are also included in this chapter and the problems associated with this category of protocols are also discussed. Even though, the delay tolerant routing protocols achieve fault tolerance in case of network partition, they do not focus on reducing the load on resource-constrained devices, because these protocols are designed for scenarios where the partition has already occurred and thus achievement of energy conservation or congestion reduction is not their priority. Most protocols in this category, with few exceptions (Pasztor et al., 2007), assume devices with sufficient resources that are capable of storing undelivered packets until the packets are transferred to devices that have a better chance of delivery to the destination.

The routing protocols that redirect the traffic towards more capable devices in order to reach the destination comes close to solving the scenario presented in this thesis. These protocols are also developed for WSNs, however, they rely on traditional routing protocols and use route request flooding techniques to not only find a route to the

destination, but to also find a more capable device in the network that can be used to redirect traffic away from the less capable nodes. This chapter also discusses the short-comings associated with the existing work in this category of routing protocols.

The chapter concludes by summarizing the problems associated with the existing protocols and identifies the gap that this thesis addresses.

2.1 Mobile Device Discovery Mechanisms in WSNs

Network protocols that use Mobile data collectors (MDCs) to transport data to its destination have implemented different discovery mechanisms. The discovery mechanisms implemented in these protocols deal with discovering the mobile devices to establish communication between the sensor nodes and the mobile devices. The majority of discovery mechanisms focus on discovering the mobile devices quickly. This section discusses the discovery mechanisms used by such networking protocols below.

2.1.1 Baseline mobile device discovery mechanisms

Somasundara et al. (2006) have targeted controlled mobility of the MDC while delivering data to the destination. In this case, the mobility of the mobile device is under the control of the sensor network and the movement of the mobile device is adjusted to improve network performance, e.g., the mobility is planned in a way that the mobile device visits all available sensor nodes deployed in the target area. They analyzed the network with and without MDCs, and report a small network energy consumption and a small increase in transmissions required to deliver data, when increasing the network density while using MDCs, without compromising the data rate significantly. The authors suggest that the sensor nodes always listen for periodic beacon packets from the mobile data collectors. Although simple, this technique of discovering mobile data collectors typically incur high energy consumption at the sensor nodes, as the sensor nodes have to always keep their radio in listening mode.

Jain et al. (2006) introduced the MULE architecture where the data generated by the sensor nodes (SNs) is delivered to the access points with Internet connectivity using MULEs. In this case, a MULE visits the sensor nodes and collects data from these sensor nodes and stores locally. The MULE then visits the access point, where it offloads all the collected data to the access point, from where the data is sent to the final destination using the Internet. A MULE (MDC) is assumed to have better resources (such as storage and energy) than the SNs, and is used to reduce work on SNs such that end-to-end multi-hop paths from the sensor nodes to the access point are not required to deliver data to the destination. In this work, the SNs listen to the periodic discovery messages broadcasted by the MULEs. However, unlike Somasundara et al. (2006), the authors of the MULE architecture use periodic listening duty cycles at the sensor nodes to discover the mobile devices, such that the sensor nodes listen for a small discovery time period (DT) every beacon interval (BI), where BI represents the time period between two consecutive beacons broadcasted by the mobile device and the listening duty cycle at the sensor nodes is given by $\frac{DT}{BI}$.

Similarly, Sadaphal and Jain (2008) analyze periodic wake up cycles of the sensor nodes. In addition, the authors of this work also analyze random wake up cycles of the sensor nodes, such that the sensor node may start listening at the radio channel at random time. The authors show that an upper bound exists for the delay between the time when the mobile device arrived in range to the time when it was detected. For both, random and periodic wakeup cycles implemented at the sensor nodes, the authors assume that the mobile devices are cooperative with the sensor nodes, such that the mobile devices broadcast periodic beacons, whereas, the responsibility of detection lies with the resource-constrained sensor nodes.

2.1.2 Using dual beacons and radio channels for discovering mobile devices

The work discussed in this section improves on the baseline discovery mechanisms for discovering mobile devices in the context of WSNs, by employing either multiple radios or multiple types of beacon in order to discover mobile devices quicker than the baseline mechanisms.

Schurgers et al. (2002) use an active polling technique to wake up a node with which data transfer is intended. The polling (beaconing) is performed on a different radio frequency than the one used for data transmission, by using two different radios on each node. The beacons contain addressing information of the source and the target node with which data transfer is intended. In this work, a node must periodically listen for beacon messages on the frequency designated for beacon packets, however no distinction is made between a sensor node and a mobile device, rather the work focuses on connectivity between two nodes of any type.

Unlike Schurgers et al. (2002), that uses two different frequency bands, Kondepu et al. (2011) target sparse WSNs and use dual beacons to detect mobile devices, typically on the same radio frequency channel. They use a long-range (LR) and short-range (SR) beacon broadcasted periodically in an interleaved manner. Sensor nodes are initially configured at a low duty cycle, which is increased as soon as they receive a LR beacon, in order to be ready to receive SR beacons. Upon reception of a SR beacon, the SN gets ready for data transmission. The authors show improved performance when compared to baseline periodic listening techniques with single beacons.

Kohvakka et al. (2009) explicitly target IEEE-802.15.4 networks assuming a hierarchical cluster-tree network topology. A node scans the channel and synchronizes with the beacons of a parent. The beacon contains synchronization information of other beaconing neighbours. The synchronization information of other beaconing nodes can be used to synchronize with a different node if the connection with the parent node fails.

This protocol reduces the scan duty cycle by using the synchronization information of non-parent devices. The protocol uses two types of beacons, i.e., network and cluster beacons. The network beacons are broadcasted a configured time before the cluster beacons. In this protocol, a node is synchronized to more than one parent at a time. The network beacon contains synchronization information about the non-parent devices, whereas, the cluster beacon contains information about the parent device. In order to receive two beacons, a child node must wake up a certain time before the cluster beacon is expected. Even though, a node does not have to always receive network beacons, but the protocol advises that the network beacons should be received sporadically to maintain synchronization information about non-parent nodes. This, however, can increase energy consumption when network beacons are received because the SN has to keep its radio longer in receive mode. In a stationary sensor network with mobile data collectors, this protocol may allow comparatively faster discovery than the baseline discovery protocols, but it does not eliminate the need for scheduling scans on all the nodes once the network is established.

The sensor nodes employing these mechanisms either need to track two different beacons or listen on two different frequencies. This can result in higher energy consumption than necessary to discover a mobile data collector, due to the additional listening operations.

2.1.3 Discovery performed by mobile devices

Liang et al. (2008) introduced low power probing (LPP) as an alternative mechanism for discovering devices in the network for data collection throughout the network. In this alternative mechanism, the responsibility of discovery lies with the gateway (destination of sensor data), such that the gateway scans the radio channel.

This mechanism of data collection is specifically designed for non-synchronized networks. In this mechanism, a gateway can periodically join the network and does not have to be always present. The sensor nodes in this mechanism store their sensor readings in

their local flash memory and broadcast a periodic probe (beacon) message. If a gateway is present, it sends an acknowledgement back to the node upon reception of a probe message. Once the node receives an acknowledgement, it wakes up and starts acknowledging periodic probes from other sensor nodes (two hops away from the gateway).

The information about all the source nodes is shared with the gateway via the sensor node(s) that is directly connected to the gateway. The gateway then requests data from each of the sensor nodes sequentially, and requests all nodes on the path to the target source node to switch to a different radio channel. This allows the other network nodes to keep probing on the original channel periodically and to keep their radios in sleep mode most of the time.

The sensor node that has requested data collection starts transmitting its data towards the gateway once the path to the gateway is established. The data packets are routed towards the gateway using a source routing protocol, e.g., DSR (Johnson and Maltz, 1996) (discussed in Section 2.2.1), because the gateway is responsible for selecting the routes. Once the data collection is complete, the gateway asks the source node to return to the original radio channel.

The authors acknowledge that the data collection from all the nodes may take a long time.

2.1.4 Learning based discovery protocols

Dyo and Mascolo (2007, 2008) and Di Francesco et al. (2010) discuss learning-based MDC discovery protocols for sparse WSNs. In these protocols, the MDCs periodically broadcast beacons, and the learning agents at less capable sensor nodes learn to reduce the listening duty cycle.

The learning agents described by Dyo and Mascolo (2007, 2008) learn to estimate the number of encounters and schedule the listening duty cycle according to the sensor node's energy budget. They divide a day in time slots and learn the temporal patterns of mobile devices arriving in their vicinity.

Di Francesco et al. (2010) use Q-learning assuming deterministic mobility, and their agents learn the probability of contact with a MDC. By doing so, they adaptively schedule the duty cycle of the SNs.

The implementation of complex learning algorithms on resource-constrained SNs is not trivial and may limit the functionalities that can be implemented on such nodes. Also, the learning agent may take a while before it can predict the contact time efficiently. During exploration time, significant amount of energy may be consumed due to activating radio channel scans when the mobile device was not present in the vicinity of the sensor node.

2.1.5 Radio triggered wake up

The most efficient methods to discover MDCs, wake up the sensor nodes (SNs) when MDC arrives in range, such that the wakeup is triggered by a special radio on the MDC (Gu and Stankovic, 2005; Ansari et al., 2009). However, such methods require extra hardware to be installed on the SNs, which may increase the size and cost of the nodes. These methods use a radio triggered wakeup circuit installed at the SN. An out-of-band wakeup signal is transmitted by the MDC to wake the SN up. The wakeup circuit at the SN utilizes the energy dissipated by the signal to trigger an interrupt at the microcontroller upon receiving this signal, without using an explicit power source of its own.

To further reduce the energy consumption at the sensor nodes, the signal used by Ansari et al. (2009) contains addressing fields, and the microcontroller at the SN goes back to sleep if the signal is not addressed for it.

2.1.6 Discussion

Overall, to the authors' knowledge, the majority of the existing protocols have placed the discovery mechanism at the resource-constrained sensor nodes, such that the sensor nodes scan the radio channel for beacons from mobile devices.

Placing the discovery mechanism on resource-constrained sensor nodes has a number of drawbacks. Implementing complex discovery mechanisms at resource-constrained sensor nodes reduces the functionalities that can be implemented on sensor nodes, due to limited memory space available on such nodes. Performing scan operation at the resource-constrained sensor nodes increases the listening duty cycles at the sensor nodes and to quickly discover mobile devices, the scan operation may have to be frequently performed. As a result the energy efficiency reduces at the sensor nodes.

A significant number of resource-constrained sensor nodes are single threaded (e.g., TelosB), i.e., only a single task can be performed at a time. Multiple functionalities scheduled on such devices may interfere with each other such that one function may get pre-empted if another scheduled function is triggered, resulting in unexpected results. For example, if a sensor node has to transmit a periodic beacon while the radio channel scan is under way, then all nodes in the vicinity may not be discovered due to the interrupt caused by the transmission of the periodic beacon, or the scan process may stop altogether due to the interruption. For this reason, just before starting the radio channel scan, all other operations have to be suspended, including periodic beacons, until the scan is completed on these devices. If the discovery mechanism is placed at a resource-constrained sensor node such that other sensor nodes depend on its periodic beacons to maintain network connectivity, then suspending periodic beacons while performing radio channel scan to discover mobile devices can result in connectivity loss.

Discovery mechanisms that assume multiple mobile devices broadcasting periodic beacons may also suffer from collisions of beacons from multiple mobile devices, resulting in failure of their discovery. In such cases, beacons from multiple mobile devices may collide with each other if the beacons are transmitted at the same time, even in the presence of CSMA/CA mechanism, because the radio transmission in this mechanism is only backed off if transmission of another device is detected.

Overall, none of the existing discovery mechanisms reduce the time taken to discover a mobile device and remain energy efficient at the resource-constrained sensor nodes at

the same time, while limiting the resource usage and implementation complexity at the sensor nodes.

In order to reduce communication load on resource-constrained sensor nodes, inspired by LPP mechanism described by Liang et al. (2008), this thesis approached the discovery problem from the inverse perspective, such that the discovery mechanism is placed at more capable mobile devices. The discovery mechanism presented in this thesis extends the LPP mechanism to utilize location information of the sensor nodes to trigger discovery mechanism at the mobile device when the mobile device expects sensor nodes in its vicinity. It is envisaged that such a discovery mechanism will not only liberate resource-constrained sensor nodes from discovering and tracking mobile devices but will also simultaneously reduce discovery interval and energy consumption at the sensor nodes. Such a discovery mechanism combined with an intelligent sensor node selection mechanism to achieve certain objectives is expected to result in efficient reduction of communication and processing load on resource-constrained sensor nodes.

2.2 Routing Protocols in Mobile Ad hoc and Wireless Sensor Networks

Typically, establishing an efficient route from a source to the destination using mobile devices is a challenging task. The paths established via mobile devices are not always available, because the path to the destination breaks as the mobile device(s), in the path, moves out of range of the remaining devices.

In resource-constrained WSNs, the problem of establishing a path to the destination becomes particularly challenging. In such networks, not only multi-hop paths have to be established from multiple sources to the destination, but those paths also have to be established in a way that allows the sensor nodes to operate for a long time. In particular, establishing energy efficient and reliable paths to the destination take a high priority in WSNs. However, the two requirements of energy efficiency and reliability

may contradict with each other such that energy efficient paths may not be reliable and vice versa. Due to the nature of WSNs, some sensor nodes may experience higher data in-flow resulting in network congestion, due to their logical or physical proximity with the destination (Somasundara et al., 2006).

Section 2.2.1 first discusses the traditional approaches to routing data from source devices to the destination, that establishes the basis of routing protocols in mobile and wireless ad hoc networks. Geographic routing protocols, discussed in Section 2.2.2, overcome the shortcomings of the traditional routing approaches, but have their own limitations. Section 2.2.3 discusses routing protocols that use link quality estimation. Such approach can be incorporated in the future work of the STERIOD protocol presented in this thesis.

Next, Section 2.2.4 also discusses routing protocols specifically designed for WSNs that incorporate mobile devices in some capacity. In addition, hierarchical routing protocols (Section 2.2.5) that resemble the network topology assumed in this thesis are also discussed to explore the possibility that these protocols can be used to offload data communication from resource-constrained sensor nodes to more capable mobile devices. Section 2.2.6 includes discussion on a few delay-tolerant routing protocols that provide a solution for routing data in intermittently connected networks.

Because the requirements set out in Chapter 1, Section 1.5 are not met by the routing protocols that employ mobile devices, Section 2.2.7 discusses device-aware routing protocols that come close to redirecting traffic towards more capable devices present in the network. However, the discussion carried out for these protocols demonstrate that such protocols also lack in achieving the goals established in this thesis.

2.2.1 Traditional Wireless Ad Hoc Network Protocols

Traditional (mobile and wireless sensor) ad hoc network routing algorithms, e.g., Perkins and Royer (1999) and Johnson and Maltz (1996), rely on route discovery procedures to build multi-hop paths toward the destination node. In doing so, these protocols flood the

network with route request (*RREQ*) messages and when a RREQ reaches the destination, a route reply (*RREP*) message is initiated by the destination. Based on these RREQ and RREP messages, paths from the source to the destination are built.

In particular, the DSR protocol (Johnson and Maltz, 1996) builds route towards the destination based on the number of hops and selects the shortest path to reach the destination. After performing route discovery, the DSR protocol stores the complete route towards the destination on the source node. The DSR protocol includes the complete route towards the destination in the data packets. The nodes, along the path, use the pre-computed routing instructions available in the data packet to select the next hop in order to deliver the packet to the destination. This (source routing) technique, however, increases the information maintained on the source node in addition to increasing the packet size.

Similarly, AODV (Perkins and Royer, 1999) also builds route towards the destination by selecting the shortest route, based on the number of hops. However, unlike DSR, for a given destination, the AODV protocol stores the next hop neighbour that is closest to the destination, on every node along the path. This technique enables AODV to take decisions in a distributed fashion while only maintaining the necessary information on each node. In addition, this technique allows AODV to reduce the size of the packet by only including the address of the next hop instead of storing the complete route to the destination.

In the case where a path breaks, a special route reply (AODV) or a route error (*RErr*) (DSR) message is propagated towards the source node, which in turn initiates a new RREQ. In DSR, if a route to the destination is known by an intermediate node, that node can send a RREP to the source without broadcasting the RREQ further.

AODV and DSR protocols were designed for MANETs, where the packets were to be delivered to a single destination from a single source node. However, typically in WSNs, multiple source nodes wish to deliver their sensor readings to a single destination (the sink). Directed Diffusion (Intanagonwiwat et al., 2000) was explicitly designed for this

scenario, and thus offers a better solution for data collection in WSNs. In this protocol, rather than the source nodes flood the network to locate a route from each source node to the destination (sink), the sink node floods the network with a data query for a particular two-dimensional area. This inverse approach reduces the overall number of requests flooded in the network, among the traditional approaches.

When the query arrives at the sensor (source) nodes in the target region, they start transmitting sensor data along the reverse paths from which the original query arrived. This means that the data is sent towards the sink through multiple paths, resulting in multiple copies of the same sensor data. When the data starts arriving at the sink node, it adjusts paths according to the arrival rate of data from different paths, re-inforcing or favouring a particular path by requesting rest of the paths to reduce their data rate.

Discussion

All protocols in this category rely on route discovery procedures that flood the network with control messages to form routes from the source to the destination. AODV and DSR flood the network with route requests and route replies when a path breaks (due to mobility), whereas, Directed Diffusion floods the network with data requests whenever the sink moves. These approaches, however, introduce high control overhead, particularly in highly dynamic environments.

Even though these protocols target routing packets in ad hoc networks, these protocols are mostly suited for stationary networks where the connections among nodes and the resulting paths to the destination do not change often. These protocols are typically viable in the environments where mobility does not frequently affect the connectivity of the nodes. In highly dynamic environments, where the nodes either frequently move or appear and disappear, resulting in changes in network connectivity and topology, these traditional protocols will incur high control overhead due to frequent network flooding with route discovery, repair and data request messages. As a result, overall network resource usage increases, i.e., quick depletion of energy reserves and congestion throughout

the network (He et al., 2003).

Such drawbacks are also discussed by relatively modern protocols, e.g., SPEED (He et al., 2003) and EARM (Akkaya and Younis, 2004) especially in scenarios with mobile nodes. EARM shows that reducing route discovery results in better energy conservation, and SPEED shows that geographic routing protocols have less energy consumption and end-to-end delays, because of the absence of route discovery phases.

Table 2.1 summarizes the characteristics of the protocols discussed in this section against the requirements set out in Chapter 1, Section 1.5 that are necessary to obtain a solution that can offload communication load away from resource-constrained sensor nodes. Table 2.1 shows that the traditional routing protocols typically only consider end-to-end latency as the influencing factor in the protocol design, by reducing the number of hops traversed while delivering packets to the destination.

Thus, any protocol that relies on flooding the network with route discovery and repair messages is unsuitable for highly dynamic environments where the paths to the destination may change often.

Requirements	AODV	DSR	Directed Diffusion
Latency Reduction	•	•	•
Device Aware	×	×	×
Energy Conservation	×	×	×
Congestion Reduction	×	×	◦
Fault Tolerance	×	×	×
Fast Reactivity	×	×	×
Communication Load Reduction at sensor nodes	×	×	×

Table 2.1: Comparison of traditional routing protocols with envisaged goals and requirements.

Legend: • = Fully Supported, ◦ = Not Fully Supported, × = Not supported.

2.2.2 Geographic Forwarding

The overhead imposed by route discovery mechanisms, in the form of route request and route reply messages, is significantly reduced by energy-aware and stateless geographic routing protocols, e.g., Karp and Kung (2000), Yu et al. (2001), He et al. (2003) and Roosta and Menzo (2004). Because these protocols build routes based on geographic location of the destination, they do not need to flood the network with route discovery messages and thus are more efficient than the traditional routing protocols.

The geographic routing protocols assume that the position of the destination node is known by the source node. Also, each node is assumed to be aware of its own location in addition to the locations of its one-hop neighbours. This class of protocols employs greedy forwarding towards the destination, generally preferring the nodes nearest to the destination. These protocols are stateless i.e., they do not need to maintain routing tables.

These protocols mainly differ in the manner in which they deal with *voids* or *local maxima*. A void or local maxima is identified as an area in which none of the neighbouring nodes is nearer to the destination than the node that currently holds the packet. For example, GPSR (Karp and Kung, 2000) employs planar graphs and the *right hand rule* to route around the void by routing the packets along the perimeter of this graph. This means that if a void is detected, then the source node will forward the packet to a node that is sequentially located in the counter clockwise direction of the source node.

Yu et al. (2001) utilises GEAR's energy-aware metric to deal with the problem of voids by minimising the estimated cost of routing towards the destination. This protocol changes the next-hop towards the destination when the remaining energy of the desired next-hop falls below the remaining energy of any of its neighbours, in turn balancing the load. When the data reaches any node within the target region, this node divides the target area into four quadrants and sends the data to one node in each quadrant. This process is repeated recursively, until the data reaches all the nodes in the target area

originally specified in the packet, i.e., until no neighbouring node, present in the target area, is found such that it has not already received the packet.

SPEED (He et al., 2003) tries to maintain a desired data delivery speed towards the destination. If a given node cannot meet the desired delivery speed, it tells the previous node to use some other neighbouring node when the next packet has to be sent to the same destination area. This technique of using feedback loops, allows SPEED to route packets around the void. In doing so, SPEED also manages to route packets around congested areas in the network. It provides three types of packet transmissions: unicast, multicast and anycast. In unicast messages, the destination is identified by *Global_ID*. This protocol floods the whole target area with data packet in order to reach the destination node (sink) identified by a *Global_ID*. A multicast message is delivered to all nodes in a certain area, whereas an anycast message is delivered to any node in a certain area. Thus *unicast* and *multicast* messages are flooded in the target area, whereas an '*anycast*' message is not forwarded any more as soon as it reaches a node inside the target area.

Roosta and Menzo (2004) select next-hop nodes geographically and probabilistically in the PGR protocol. Neighbouring nodes towards the destination are selected from the subset of nodes that are within an angle θ of the destination (Figure 2.1). The angle θ is expanded until at least two neighbouring nodes are found but never beyond 180 degrees. This limitation of expanding the angle only up to 180 degrees guarantees that the packet will not be forwarded in the opposite direction. The next-hop is selected probabilistically from the subset of the neighbouring nodes which are within an angle θ from the destination node. The probability of the next-hop node selection depends on its remaining energy and the number of retransmissions required to reach the next hop, which is considered to be the inverse of the backward link reliability (signal strength from next hop to the source). This protocol might fail to route around a void due to the limitation of not extending the angle beyond 180 degrees.

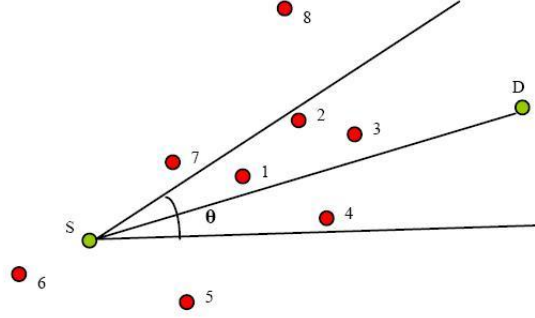


Fig. 2.1: PGR example, illustrated from (Roosta and Menzo, 2004)

Discussion

Geographic routing protocols offer the best solution to reduce control overhead, especially in energy-constrained networks such as WSNs. From application's perspective, these protocols are best suited to deliver data queries to a certain geographic area.

Generally, delivering data from the source nodes back to the sink node is not as efficient. Ideally this data should be delivered to only one node usually identified by some *identifier* (ID), e.g., address. However as can be seen from the above discussion, this is not the case.

This problem of delivering data to the sink becomes more evident in cases where the sinks are mobile and change their location. To explain the problem further, consider a scenario where a mobile sink issued a data query for a particular target region. The sensor nodes in the target region, upon reception of the query, start sensing data and try to send data, at a rate possibly specified in the query, towards the sink using the sink's location, which might have been specified in the query itself, using the geographic forwarding protocol in question. However, when the data arrives at the location of the sink, the sink (being mobile) has already moved to a different location. Hence, in such a scenario, a geographic routing protocol (if employed alone) completely fails to deliver the requested data to the sink. This is also evident from the discussion carried out in Section 2.2.4.

Geographic routing protocols are best suited for cases where the packet has to be sent to a stationary sink node whose location is known by the source nodes or when the packet has to be delivered to a certain geographic region. It might not be feasible to pre-configure or disseminate the location of all nodes in the network to all the other nodes due to memory limitations, especially in cases where any node may need to send data to any other node, e.g., in a mesh network topology and especially in large-scale dense deployments.

If a node is in an IP network, as is the case for the command and control server in the model presented in this thesis, the geographic location of such devices might be irrelevant, because such devices may be located in data centres possibly in a different country or continent. In such case, geographic routing protocols can only deliver data to a gateway that is co-located within the WSN, from where the packet is forwarded to the CC via IP network.

Since these protocols follow greedy geographic forwarding towards the target location, and prefer neighbours which are the closest to the destination, they do not take advantage of the diverse types of devices available in the urban environments. For example, consider a device that wishes to send data to the target region. The closest neighbour to the destination may be a less capable device such as a *sensor node*, however more capable devices such as *mobile devices* may be available. These protocols will prefer the *sensor node* over the more capable *mobile device* in such a scenario, because the *sensor node* is closest to the destination, thus being unable to take advantage of more capable devices present in the vicinity.

From the above discussion, we can see that these protocols, as they are currently designed, may not be the best approach for routing packets towards mobile sinks and gateways, in order to take advantage of paths created via more capable mobile devices. In such cases, if feasible, the packet might have to be forwarded to devices that are further away from the destination, so that more capable devices could be utilised. In addition, the mobile devices will be considered as destinations instead of the command and control

Requirements	GPSR	GEAR	SPEED	PGR
Latency Reduction	•	•	•	•
Device Aware	×	×	×	×
Energy Conservation	×	•	×	•
Congestion Reduction	×	×	•	×
Fault Tolerance	•	•	•	×
Fast Reactivity	•	•	•	•
Communication Load Reduction at sensor nodes	×	×	×	×

Table 2.2: Comparison of geographic routing protocols with envisaged goals and requirements.

Legend: • = Fully Supported, ◦ = Not Fully Supported, × = Not supported.

server in order to apply geographic forwarding, as is the case in protocols discussed by Ye et al. (2002) and Kim et al. (2003). The geographic routing protocols have partially influenced the work presented in this thesis in the way that geographic localization information is used by the mobile devices to perform discovery of sensor nodes in the ‘Inverse Location-Aware Discovery’ mechanism and to determine the potential children of a particular sensor node in the STERIOD protocol.

Table 2.2 summarizes the characteristics of the protocols discussed in this section against the requirements set out in Chapter 1, Section 1.5 that are necessary to obtain a solution that can offload communication load away from resource-constrained sensor nodes. All the protocols discussed in this section can react fast to the changing environment due to their stateless nature, and reduces end-to-end latency by utilizing the nodes that are the closest to the destination. Only GEAR and PGR use energy-aware metric to reduce energy consumption, while only SPEED reduces congestion by maintaining a desired delivery speed to the destination. Apart from PGR, all the protocols discussed in this section are fault tolerant because they are capable of routing around

disjoint portions of the network (voids). However, none of the protocols discussed in this section are device aware and thus cannot implicitly reduce communication load on resource-constrained nodes because they naturally select the node that is closest to the destination irrespective of its capabilities.

2.2.3 Routing with Link Estimation

Different protocols take routing decisions based on different factors, e.g., the protocols discussed in Section 2.2.2 take packet routing decisions mainly based on the next hop's physical proximity to the destination, whereas, the protocols discussed in Section 2.2.1 take routing decisions based on the number of hops between the source and the destination via different routes while reducing the average number of hops that a packet traverses.

Another interesting trend in the research around routing protocols is to take decisions based on the link quality of the next hop node, possibly combined with some other factors such as energy reserves. This section briefly discusses the traits of a few protocols that use link quality estimation technique in some capacity.

Gnawali et al. (2009) presented the collection tree protocol (CTP) that creates a cluster tree network by using estimated link cost to the next hop and adaptive beaconing techniques. This protocol is particularly interesting as the network formation mechanism in the STERIOD protocol (presented in this thesis) is partially influenced by the ideas presented by Gnawali et al. (2009). The CTP protocol forms the network such that the collection point of the data packets advertises a cost of zero, and this cost gradually increases per hop. The main factor that establishes the cost in this protocol is the estimated number of transmissions to successfully deliver data to the next hop. This routing protocol tries to reduce the number of transmissions while forwarding data towards the collection point. The authors use adaptive beaconing technique that adjusts the beacon transmission rate according to the environment, such that the interval is increased if the links remain stable upto a maximum limit. The beacon interval is reset

to its initial value if a neighbouring node requests a beacon, a sibling node (at the same level of the tree) requests data to be forwarded or if a node's cost reduces significantly.

Woo et al. (2003) argue that instantaneous value of the received signal strength is a poor indicator of the link quality as the quality of even stable links change over time. For this reason, the authors use window mean with exponentially weighted moving average to estimate the average success rate of packet delivery over a time period. The authors also argue that complex techniques such as linear regression and kalman filters are possibly impractical to be implemented in resource-constrained environments such as sensor networks. The authors also control the transmission power in order to conserve energy on the sensor nodes. A similar technique is also applied in the research carried out in this thesis.

Discussion

The protocols discussed in this section argue that instantaneous values of signal strength traces are poor indicators of a link's actual quality. For this reason, the protocols generally use averages gathered over a period of time. The collection tree protocol performs congestion control at the channel by adaptively adjusting the beacon interval of the sensor nodes. Similarly, Woo et al. (2003) can also achieve some reduction in congestion due to their estimator that favours links with better average link quality.

As can be seen from Table 2.3, the protocols discussed in this section do not fulfil the requirements set out in Chapter 1, Section 1.5, however, they provide interesting basis for building a reliable protocol that can fulfil the requirements set out in this thesis.

The collection tree protocol (CTP), along with the EMMON project (Tennina et al., 2011a), in particular influenced the design of the network formation mechanism in this research, in terms of the network topology and broadcasting forwarding cost in the beacon packets. However, while forming a network, link estimators are not taken into account in this research as the main focus was to take advantage of more capable mobile devices present in the vicinity of the sensor network. The adaptive beaconing mechanism

Requirements	CTP	WMEWMA
Latency Reduction	•	•
Device Aware	×	×
Energy Conservation	×	×
Congestion Reduction	•	◦
Fault Tolerance	×	×
Fast Reactivity	×	×
Communication Load Reduction at sensor nodes	×	×

Table 2.3: Comparison of protocols that use link quality estimation.

Legend: • = Fully Supported, ◦ = Not Fully Supported, × = Not supported.

presented in CTP is not used in this research in order to allow mobile devices to be able to discover beaconing sensor nodes and offload communication load away from the resource-constrained sensor nodes. This is due to the reason that if the beacon interval changes then the change in the beacon intervals must be communicated with the mobile devices so that the mobile devices can then successfully discover sensor nodes, thus increasing the communication within the sensor network.

The work presented by Gnawali et al. (2009) and Woo et al. (2003) will potentially influence the future work of this research such that average link quality estimates that are received from the beacons of the neighbouring devices can be used during the network formation phase. However, to make better informed decisions, the link quality estimates will have to be averaged over a comparatively higher number of beacons to be meaningful (Woo et al., 2003). This might lead to an increase of the network formation and post link-loss resynchronization times. This time delay renders the use of link quality estimation unsuitable for mobile devices, that might be present only for a short time, wishing to communicate with the WSN.

Therefore, the work presented in this thesis does not utilize link estimates while

routing, but uses the risk of data loss per device type while routing data packets towards the destination instead, as discussed later in Chapter 3 under Section 3.4.2 and shown in Equation (3.17), as a form of generic link quality estimation for devices of the same type.

2.2.4 WSNs with Mobile Devices

Protocols reviewed in this category use mobile devices in some capacity within the context of WSNs.

The majority of the protocols that employ mobile devices, use different mechanisms to track the mobile devices in order to deliver sensor data to them. Such protocols use mobile devices as sinks, such that the sensor data is consumed by these mobile devices and the sensor network must deliver data to individual mobile devices that have requested the sensor data from the sensor network.

Other protocols in this category use mobile devices as intermediary nodes to forward packets towards the destination. Protocols that take advantage of more capable (mobile) devices are discussed in Section 2.2.7.

Because the envisaged goal of this thesis is to take advantage of mobile devices while delivering sensor data to the destination, the protocols in this category are considered to be closely related to this research and therefore, are discussed in detail.

2.2.4.1 Two Tier Data Dissemination (TTDD)

Ye et al. (2002) introduced TTDD, in which they construct logical grids as overlays in the WSNs in order to transport data from the WSN nodes to mobile sinks and reduce control overhead incurred by the sink's mobility. Such a grid is shown in Figure 2.2. The authors assume that all nodes in the WSN know their location in addition to their mission a priori, e.g., tracking a hostile target or sensing environmental phenomenon such as temperature. This protocol is based on greedy geographic forwarding and can use any geographic routing protocol as its underlying routing protocol, e.g., Karp and

Kung (2000) or Yu et al. (2001). The protocol identifies two nodes in the WSN that act as a proxy for each mobile sink and hides the mobility of the sink from the other sensor nodes in the network.

Typically in sensor networks, the sensor data is reported to the destination in one of three ways: via periodic reporting, generating alerts when the phenomenon is observed, as a response to a query (Tennina et al., 2011a). The TTDD protocol takes the alert based approach, i.e., the sensor data is reported when the phenomenon is observed based on some pre-defined conditions.

When the phenomenon (stimulus) is observed, all the sensor nodes surrounding the phenomenon collaborate with each other to aggregate data, and one of the nodes starts generating reports. At this point, this source node assumes itself to be at one crosspoint of the intended logical grid ('B' in Figure 2.2), and calculates the locations of its four neighbouring dissemination crosspoints given its own location and the cell size ' α ', where α is a configuration parameter. The source node then sends a *data announcement* message for each of its four neighbouring dissemination crosspoints towards that crosspoint. Each of these *data announcement* messages reaches near the intended dissemination crosspoint through a multi-hop path using geographic forwarding. The node nearest to an intended dissemination crosspoint location assumes the role of the dissemination crosspoint node, since the deployment of sensor nodes may not follow a perfect grid structure. This node then forwards the *data announcement* message in a similar fashion to its four neighbouring dissemination crosspoint nodes, except the one from where it received it, recursively creating a grid throughout the WSN. In this way every source node that has observed the phenomenon, creates a virtual grid for itself throughout the WSN.

When a mobile sink requires data from the sensor field, it floods the *query* (depicting its interest in the sensor data) locally within the area of about cell size ' α ' (mobile sink 'S' flooding a query in radius ' α ' is depicted in Figure 2.2). Once the locally-flooded query initiated by the mobile sink reaches a dissemination node, called the *immediate*

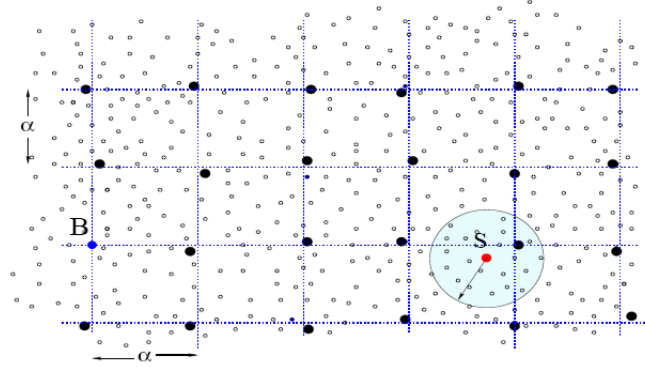


Fig. 2.2: TTDD's logical grid showing a source 'B' and a sink 'S' (Ye et al., 2002)

dissemination node, the query is then forwarded upstream towards the source by this dissemination node, over the grid. From the query forwarding, the upstream dissemination nodes store the locations of the downstream dissemination nodes, constituting routes towards the mobile sink, or more precisely, towards the *immediate dissemination node*. If a dissemination node receives queries from more than one downstream dissemination node for the (same) data to be acquired from the same source, then it only forwards a single query towards the source, resulting in path sharing and reduced traffic.

Each mobile sink associates itself to two sensor nodes called the *primary agent* and the *immediate agent*. The primary agent is used as a proxy for the mobile sink and makes the sink's movement transparent to the rest of the WSN. The data from the source flows towards the primary agent along the virtual grid. Once a sink is about to move out of range of its primary agent, it selects the sensor node closest to itself as an immediate agent, and shares the location of the immediate agent with the primary agent. If the sink is about to move out of range of its current immediate agent, then it selects a new immediate agent and shares its location with the primary agent via the old immediate agent. The communication between primary and immediate agents follow greedy geographic forwarding. If the sink moves farther than a cell size ' α ' from the primary agent, it selects a new primary agent and floods the data query again, within

this cell, to find a new immediate dissemination node.

Discussion Since the mission of the WSN is assumed to be known a priori, the authors suggest that if the mission changes, then the intended new mission has to be flooded throughout the WSN. This means that the WSN can be used by only one type of application at a time. The authors analytically show that their approach has a lower worst case overhead and state complexity than the Sink Oriented Data Dissemination schemes that flood the network with the location of the sink whenever the sink moves, such as Directed Diffusion (Intanagonwiwat et al., 2000), while having comparable performance in the stationary sink case.

As a separate grid is constructed for each source, the overhead of routing towards a mobile sink that could possibly be interested in receiving data from more than one source increases proportionally. In addition, in comparison to the scenario and envisaged goals in this thesis, this protocol does not leverage mobile devices to reduce network load within the sensor network.

Because the packets travel along the grid, the nodes present along the grid or cell boundaries may witness more energy depletion than nodes that are inside a cell and do not lie along the cell boundaries. Due to this, the traffic load may not be balanced across the whole WSN.

2.2.4.2 Scalable Energy-efficient Asynchronous Dissemination (SEAD)

The SEAD protocol also addresses the problem of delivering sensor data to multiple mobile sinks by sharing paths within the sensor network. Kim et al. (2003) presented a minimum-cost weighted Steiner tree based approach, to routing data towards multiple mobile sinks from WSN nodes. A Steiner tree can be defined as a tree that connects N points by lines of minimum total length in a way that any two points may be connected either directly or indirectly via other shared points. Figure 2.3 shows a Steiner tree to service multiple mobile sinks by sharing the paths among them via shared points called

replicas.

The SEAD protocol also uses greedy geographic forwarding as the underlying routing protocol, and builds a set of Steiner trees as an overlay towards mobile sinks in order to share routes towards multiple mobile sinks if possible. It creates one Steiner tree per source node. Just like TTDD, the nodes surrounding the stimulus collaborate with each other to aggregate data, so that one node can act as a source node. The source node is the root of the tree and sends its data to multiple mobile sinks along the tree. This is done to reduce the traffic when multiple mobile sinks request the same or similar data from the same source. This protocol identifies a sensor node in the WSN to act as a proxy for the mobile sink and extends the path from this proxy node to the mobile sink as the mobile sink moves in the WSN's vicinity.

Initially at bootstrapping time, the stationary sensor nodes in the WSN, as well as the mobile sinks, discover their one-hop neighbours. The mobile sink identifies its nearest node as an *access node*, and sends a *join query* to the access node, which in turn forwards this join query, containing the location of the access node with the desired update rate, to the source node using the underlying (geographic) routing protocol.

The source node, being the root of the Steiner tree, then selects children nodes among the neighbouring nodes that lie towards the access node. The selection of children nodes is based on their proximity to the access node and the desired update rate requested by the access node. This is done so that the paths to multiple mobile sinks can be shared. In doing so, the protocol identifies intermediate destinations, called replicas such that the path is shared by multiple mobile sinks from the common source, after which the Steiner tree will be divided into branches (Figure 2.3). This function occurs recursively until the source is connected to the access node. This recursive search among the child nodes of a given node is performed to find appropriate replica points, such that if possible multiple mobile sinks can be served through the same dissemination tree branch.

As the mobile sink moves away from its access node, it selects the nearest node and sends a *path setup* message to the access node, using this nearest node. All of the nearest

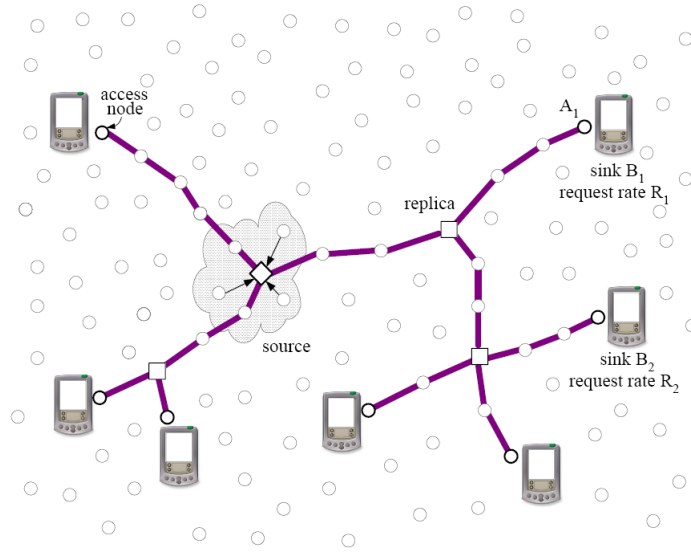


Fig. 2.3: SEAD's steiner tree to service multiple mobile sinks (Kim et al., 2003)

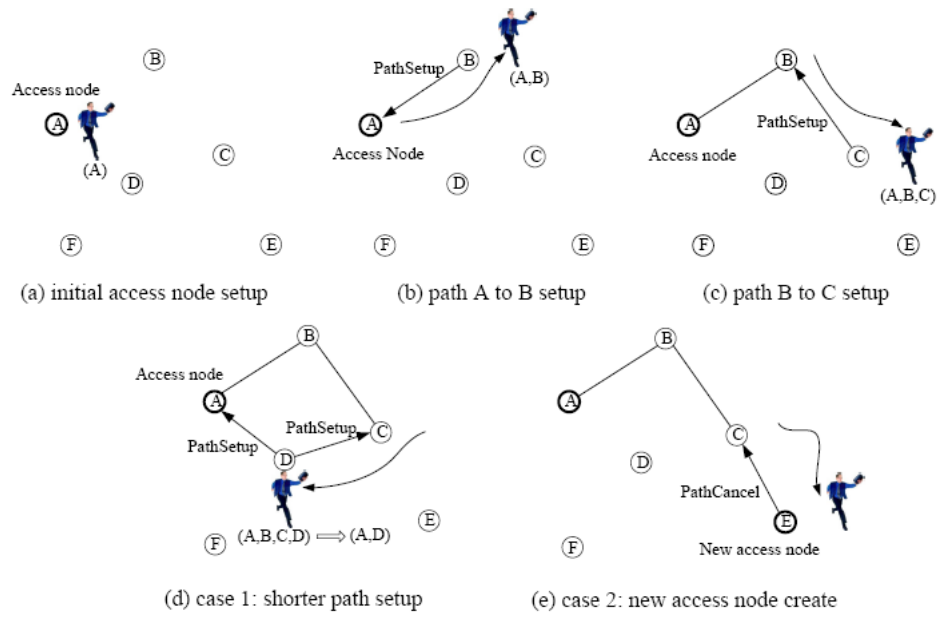


Fig. 2.4: Construction of path from access node to the mobile sink (Kim et al., 2003)

nodes identified by the mobile sink, while moving in any direction, become part of the forwarding path. If a new nearest node is closer to some earlier node in the forwarding path, then the sink asks the latest nearest node to setup a direct path with the earlier node present in the forwarding path. This is done, so that shorter paths could be used, if present. This process is shown in Figure 2.4, from steps (a) to (d).

If the hop count between the sink and its access node increases beyond a predetermined threshold then the sink replaces the old access node with a new access node, as shown in Figure 2.4, step (e). The authors claim that the cost of changing the access node is higher than forwarding data a few additional hops, due to the control overhead, so they keep the threshold value high at the expense of increasing end-to-end delay. If the difference between the old and new access nodes is less than another predetermined threshold, then the new access node is connected to the old replica in order to minimise tree construction overhead, otherwise the sink's position is sent to the source and new replicas are calculated.

Discussion The authors evaluate their approach against TTDD and Directed Diffusion, and report a smaller energy consumption per node on average than either of these two protocols. Because their evaluation criteria averages the energy consumption over all the nodes in the network, and the paths are shared by multiple mobile sinks, this means, however, that nodes that are on the branches of the Steiner tree will experience higher energy consumption than other nodes. This behaviour can be seen in Figure 2.3. This results in some nodes' energy being depleted more quickly than others, and the traffic load not being shared fairly among all the nodes in the WSN. The authors also report that their approach has lower average delay than TTDD, but higher than Directed Diffusion, since their approach finds shared paths which might be less optimal than the paths used in Directed Diffusion.

SEAD also aims to support only mobile sinks such that resource-constrained sensor nodes strive to build reliable paths towards the mobile devices and does not leverage

mobile devices to reduce load on resource-constrained sensor nodes. For this reason, this approach is not suitable either in the scenario presented in this thesis, where the paths that could potentially provide better end-to-end delay, improve energy consumption of the WSN nodes and reduce congestion at critical nodes, can be leveraged. Just as in TTDD, SEAD also constructs Steiner trees separately for each source, therefore, multiple steiner trees are built in the network, one for each source node in the network.

2.2.4.3 Energy Aware Routing To Mobile Devices (EARM)

Akkaya and Younis (2004) also focus on routing data from WSN to mobile devices (sinks). Their protocol can be based on any traditional MANET routing algorithm, e.g., Perkins and Royer (1999) or Johnson and Maltz (1996), where the authors try to minimise route discovery overhead by extending the paths from the source node to the mobile sink. However, unlike TTDD and SEAD, the EARM protocol does not share paths when multiple mobile sinks are present in the vicinity of the WSN.

Initially EARM performs route discovery from the stationary source nodes in the WSN to reach the mobile sink, which is in the vicinity of WSN, using the underlying routing protocol. The protocol assumes that the mobile sink moves in strides, possibly pausing for some time before initiating the next stride. This protocol also extends paths towards mobile sinks, based on the path discovered initially through route discovery, when necessary.

The authors use a three-step process to extend the lifetime of the currently discovered routes in order to continuously deliver data to the mobile sink. When the initial route discovery is performed, the protocol identifies a relay node that can communicate directly with the mobile sink. Once the mobile sink is about to move out of transmission range of the relay, the mobile sink (probably predicting its motion) instructs the relay node to increase its transmission power, in the hope that the relay node can still deliver data to the mobile sink in a single hop. If increasing transmission power is no longer feasible, the mobile sink identifies a forwarder node (nearest to itself) through which it can reach

the relay node. This step increases the number of hops between the mobile sink and the relay node, which was initially communicating directly to the mobile sink, resulting in the extension of routes from the source nodes to the mobile sink. This step of increasing the number of hops between relay node and mobile sink is repeated until it becomes inefficient to increase these hops further. As a third and final step, if increasing the number of hops from the relay node exceeds a certain threshold, the mobile sink then initiates a full route discovery procedure.

In this way, the authors reduce the control overhead incurred by route discovery, whenever the mobile sink moves away from its current location and relay node. The authors of this protocol evaluate their approach against AODV and DSR in simulations and show that their approach achieves improvements in energy consumption and performs close to the baselines in other aspects.

Discussion Although this protocol enables the traditional wireless ad hoc network routing protocols to cope with the movement of sinks, it will still incur significant control overhead if mobile devices are used for data forwarding to reduce load on the resource-constrained sensor nodes. The reason for this is that this protocol still depends on route discovery and will have to perform route discovery every time an intermediate mobile device moves away from the nodes constituting the path to the destination. Additionally, increasing the transmission power as a mobile device moves away will increase the energy consumption at the resource-constrained sensor nodes. The reasons for the unsuitability of this protocol in a highly dynamic environment are quite similar to the ones discussed in Section 2.2.1.

2.2.4.4 Intelligent Agent-based Routing (IAR)

Kim et al. (2010), discuss an agent-based approach to routing data to the mobile sink. Just like the protocols described earlier in this section, this protocol also takes a similar approach of identifying a sensor node in the WSN as a proxy for the mobile sink, and

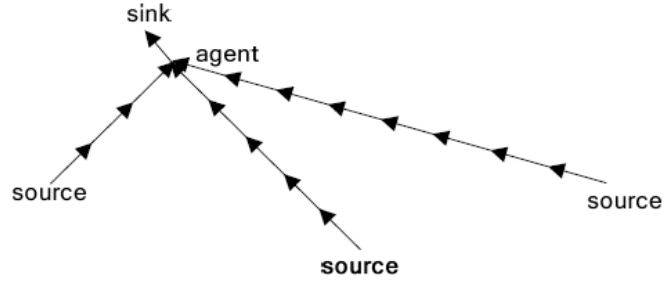
later extends the path between the proxy and the mobile sink by identifying relays. At a later stage, when the path through the proxy (agent) becomes inefficient, the protocol establishes connectivity between the mobile sink and the source without involving the proxy (agent).

The authors suggest that when a mobile sink arrives in the vicinity of a WSN, it broadcasts a *Hello Request* and all of its one hop neighbours reply with a '*Hello*' message. From this *Hello* message, the sink determines the closest node and identifies it as the agent.

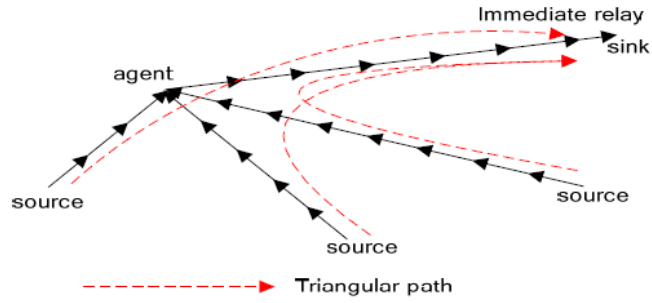
After the sink identifies its agent (proxy), it sends a query to it, which the agent forwards to its neighbours, resulting in the flooding of the query. Each sensor node, based on this query identifies the next hop towards the agent, in the reverse direction of the query, establishing gradients towards the sink. If more than one neighbour could act as a next hop towards the agent, then the neighbour from which the query was received first will be preferred. As in TTDD (Ye et al., 2002), one of the sensors surrounding the phenomenon that has been observed becomes the source and generates data reports. If one of the next hops fails, then the current node broadcasts a *path request* message to all the neighbours, each of which reply with a *path reply*, re-establishing the connectivity.

If the sink does not receive data for some predetermined time, then the sink thinks that it has moved out of the range of the agent and hence broadcasts a *relay request* message to all of its neighbours, each of which replies with a *relay reply* message. The closest node to the sink is selected as an *immediate relay* in order to reconnect the agent with the mobile sink. The sink transmits a *relay path setup* message to the agent via the immediate relay, since all nodes already know the path towards the agent from the *query* message. The agent after receiving this *relay path setup*, routes data packets towards the relay using the reverse path of the *relay path setup*, in order to reach the mobile sink. If the agent has an old relay path for the same sink, then the agent transmits a *relay path clear* message along the old relay path.

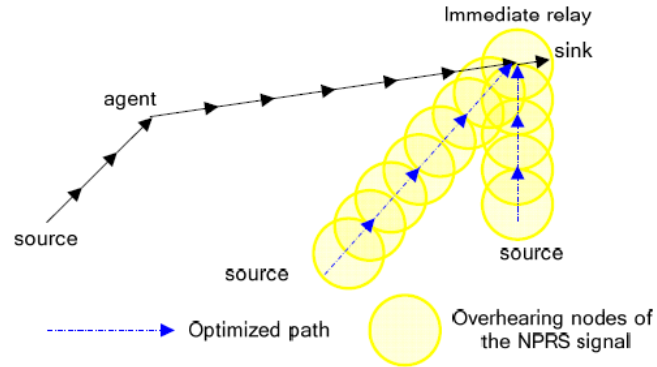
The immediate relay calculates the distance from source to agent and then to itself



(a) The path setup from source to mobile sink via agent.



(b) A mobile sink moving away and the path extended between agent and mobile sink.



(c) Path shortened after difference became significant.

Fig. 2.5: Construction of path from source node to the mobile sink. The path is shortened, once the old path becomes inefficient (Kim et al., 2010).

and calls it *old path*. It also calculates the linear distance (point to point) between the source and itself, calling it *new path*. If the new path is significantly shorter than the old path, then the immediate relay selects the new path to improve path efficiency. In this case the immediate relay transmits a *new path setup* message to the source. The source upon receiving this message, starts sending data along the reverse path of this message. Once the immediate relay receives the first data packet, it issues an *agent path clear* message to the agent, so that the agent can remove its state and the path via the agent can be discarded.

Discussion The authors compare their approach to that of Ye et al. (2002), and mathematically show that their approach incurs less control overhead. The average end-to-end delay and energy consumption are reported to be comparable, however, the authors report better throughput than TTDD. Since the query is flooded across the whole WSN initially, every node in the network creates paths toward the sink. However, this approach also results in high control overhead in terms of establishing paths. If the number of mobile sinks in the network increases, the control overhead associated with the flooding of the query along with maintaining the paths towards the mobile sinks will increase proportionally.

This protocol uses *path request* and *path reply* messages that are generated upon disconnection with the next hop. Additionally, similar techniques are also used to clear old paths from the agent node to the old relay node, as the mobile device moves and selects a new immediate relay, and while creating new paths from the source to the relay node when the paths via the agent become inefficient. These techniques can result in increased control overhead while resource-constrained sensor nodes try to keep connected to the mobile device, because of the additional transmissions of the control packets.

This protocol cannot reduce communication load and energy consumption on the resource-constrained sensor nodes while delivering data to the destination by leveraging mobile devices, because the control overhead increase with the movement of the mobile

devices is expected to offset any benefit.

2.2.4.5 Predictive QoS Routing to Mobile Sinks

Kusy et al. (2009) introduce mobility graphs for encoding knowledge about mobility patterns in the network. Unlike the protocols discussed earlier, in this protocol, the authors use training data to predict mobility and choose future relay nodes, that act as proxies for the mobile sinks and that are connected to the mobile sinks while the previous relay nodes are still active. Thus, extending the paths between the mobile sink and the proxy node, via other intermediary nodes, is not required in this protocol. The mobile nodes in this protocol only act as sinks and are not used for data forwarding to other networks, e.g., IP networks.

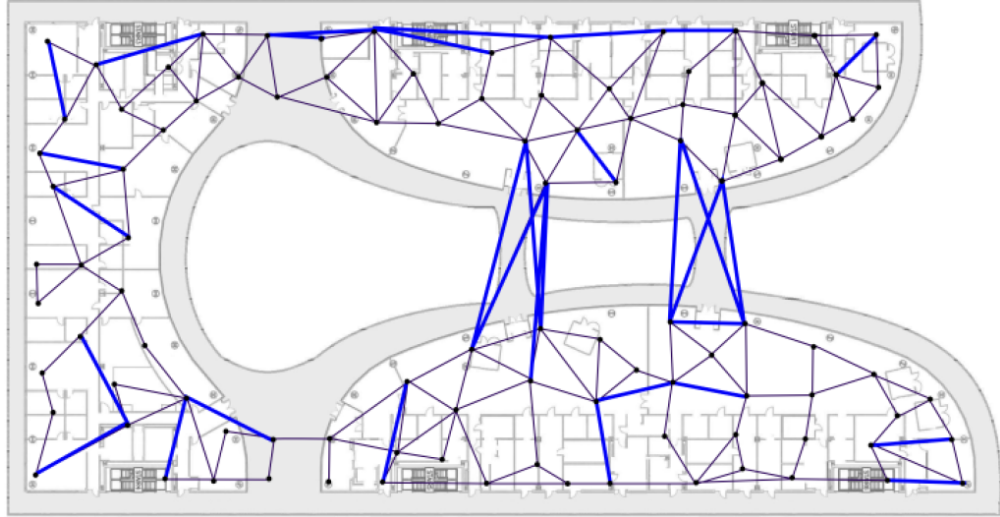
During the training period, the authors build connectivity graphs in the WSN and store them in the network. In this graph each sensor node is considered to be a vertex of the graph, and the edges between the vertices represent connectivity among different sensor nodes. Simultaneously, mobility graphs are built and stored in the network. This mobility graph is based on radio signal strength (RSSI) traces, by identifying the best connected sensor node to the mobile sink, at a given time. The RSSI traces allows the protocol to indirectly observe user's trajectories. When a mobile sink moves and its best connected sensor node changes from ν_m to ν_n , then an edge is inserted in the mobility graph to represent this fact. This mobility graph is then used to predict the trajectories of users (who are carrying mobile devices that act as mobile sinks) and the future best-connected node, to act as a relay to the mobile sink, after the training period is over. The authors use differences between connectivity and mobility graphs to identify regions where the routing information has to be pre-computed.

The authors match current RSSI traces to the traces recorded in the past, and minimize the impact of a mobile node's varying speed from the prediction (as each user may have a different walking speed). By minimizing the effect of speed, the transit time from the previous relay node to the next relay node is also pre-computed. In particular,

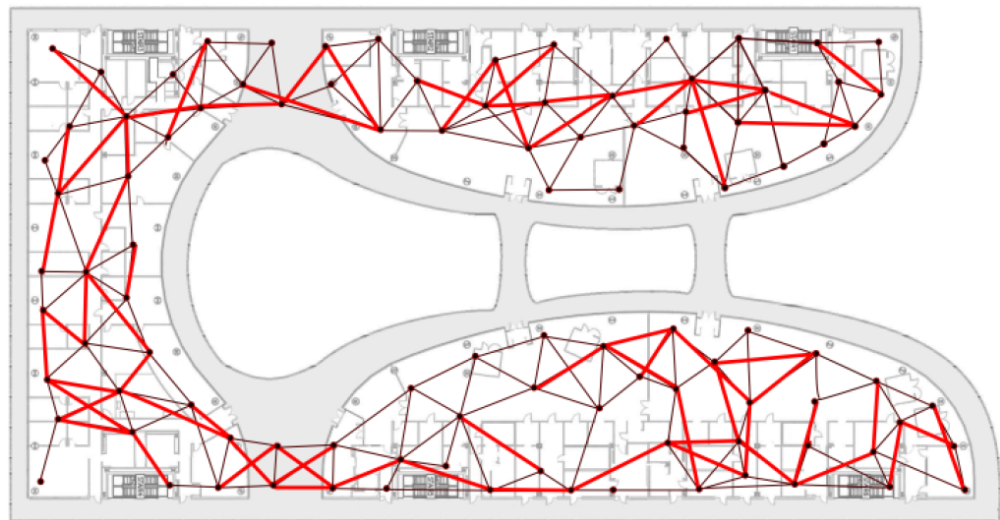
this means that when the mobile user (mobile sink) moves away from the previous relay node and towards the next relay node, the sensor nodes before the relay node can switch the delivery of packets towards the next relay node instantaneously, i.e., as soon as the next relay node becomes the best-connected node.

The authors use *information potential* based routing, where *information potential* is a real function whose value is '1' at the sink/relay and '0' at a node that can never be the relay node (0-node), at any other node the function's value is the average of its neighbours. This type of routing takes into account the robustness of the path in addition to the path length, which may lead to the selection of sub-optimal paths in terms of path length. The packet is forwarded to the neighbour with the highest *information potential* value, which induces a routing tree towards the sink/relay. Each node periodically updates its *information potential* value, averaging the values of its neighbours unless it is a sink or a 0-node.

If the position of a sink changes slowly towards a neighbouring node, then the *information potentials* are updated in only a few iterations. The information potential update begins at the relay node and propagates toward the source recursively. This information potential update is further improved by using mobility prediction information, which decreases the number of iterations required to update the information potential. Each node stores two potential values for each mobile sink, one holding the current value and the other holding a precomputed predicted value. Each node advertises its current and precomputed *information potential* value for a given sink by periodically broadcasting it along with the time to transition and the identifiers of current and predicted relay nodes. The current information potential represents the current relay node and the predicted value represents the future relay node. The protocol also stores *precomputed information potentials* for non-local movement (Figure 2.6), using the mobility and connectivity graphs. Non-local movement refers to the mobility of the mobile device between physical areas that are not directly connected (Figure 2.6a). This enables the protocol to predict where and when such movement will occur. Based on this information a simple switch



(a) Mobility graph



(b) Connectivity graph

Fig. 2.6: Difference between mobility and connectivity graphs (Kusy et al., 2009). (a) Additional lines represent represent mobility between opposite sections of the building, representing non-local movement. (b) Direct connectivity is not present between the opposite sections of the building.

to the *precomputed information potential* will occur.

Discussion The authors highlight that the communication and storage requirements grow linearly as the number of mobile users increase. However, in a network of thousands of nodes and hundreds of mobile users, it will become prohibitive to store and maintain information potentials for each mobile user. Among the protocols that deal with mobile sinks, this protocol has a smaller control overhead than the majority of protocols reviewed in this section. However, this protocol only emphasises routing data towards the mobile sink and does not leverage the mobile devices to deliver data to a distant destination while providing fault tolerance in the case of network partition, and reducing energy consumption and traffic congestion at the sensor nodes. Furthermore, as a lot of processing is performed by stationary sensor nodes to track the mobility of mobile sinks, the protocol may not be suitable for resource-constrained and battery-operated WSNs.

2.2.4.6 Routing Over Mobile Elements (ROME)

Basagni et al. (2010) introduced an approach to routing data by using mobile devices as relays towards a stationary base station or gateway, where they borrow the coloring mechanism from the ALBA-R protocol (Casari et al., 2007), associating forwarding choices to nodes depending on their color in the ROME protocol. Unlike the protocols discussed above, the ROME protocol does not address the problem of delivering data packets to a mobile destination (sink), rather it uses the mobile devices as data forwarders (relays) within the WSN.

The authors assume a stationary network of WSN nodes, with mobile devices moving in their vicinity. These mobile devices are used as relays to deliver data from sensor nodes towards the base station. In addition to being relays, the mobile devices in this protocol could also act as source nodes.

The colors range from C_0 to C_{max} , where the sensor nodes nearest to the base station

have color C_0 and the sensor nodes farthest away from the base station assume color C_{max} . A packet generated or relayed by a node having color C_k follows a route which first traverses through C_k nodes, then through C_{k-1} nodes and so on, finally reaching the base station. A mobile node is assumed to detect when it is in motion in addition to knowing its own location. The stationary sensor nodes are only assumed to know their location, and the location of the stationary base station (sink).

A node that wishes to send a packet to the base station tries to find a relay (next-hop towards the base station), by broadcasting a RTS packet containing its own location and the range of colors that the next-hop node should belong to. If the sending node does not have a color, then the range of colors that the sending node looks for in a potential next-hop node is C_0 to C_{max} , otherwise the requested range of colors is C_0 to $C_{nodeColor}$, where $C_{nodeColor}$ is the color of the sending node.

A relay node that already has a color, determines that its color belongs to the range specified by the sender in the RTS. If this is the case, then the relay broadcasts a CTS packet containing the relay's identifier, node type (mobile or stationary) and its color.

The sender, upon reception of CTS, transmits its packet to the relay in order to reach the sink (base station) and assumes the color specified in the CTS packet by the relay. However, if the sender is a mobile node and is in motion, then the sender does not assume this color and the sender's color remains *undefined*. If the sender receives CTS from more than one relay, then the range of requested colors is divided recursively and a new RTS is sent for each range, until the sending node finds an appropriate relay.

Discussion Since the mobile devices do not assume a color while in motion, a mobile node cannot act as a relay while it is in motion i.e., the mobile node can only act as a relay when it is stationary and it has acquired a color. The authors evaluate their approach against an optimised version of GPSR (Karp and Kung, 2000) and claim that their approach achieves a higher delivery rate than GPSR and consistent latency, whereas GPSR suffers higher latency with increasing traffic load.

Apart from the limitation of this protocol in terms of using mobile devices that are not in motion, the protocol cannot leverage mobile devices to deliver data to a command and control server located in an IP network. The reason being that all the nodes assume color with respect to the position of the stationary base station (sink). The mobile devices, in this protocol, assume color with respect to the (sensor) nodes in their vicinity rather than offering their neighbours with an alternative path to the command and control server, such that the nodes may assume color based on the mobile device.

Even though the authors utilize mobile devices to bridge between nodes, by design the protocol does not aim to reduce congestion at critical network nodes or to reduce communication load at the stationary resource-constrained nodes. The communication load on the sensor nodes is only reduced when the packets are forwarded via mobile devices that are not in motion. In addition, the protocol does not consider the energy consumption of the nodes as an influencing factor of its decision metrics to protect them from depleting their energy reserves.

2.2.4.7 Summary and General Discussion

The greatest similarity, in terms of network architecture, with this thesis is seen in MEMOSEN, where Chen and Ma (2006) consider a multi-tiered architecture and analyse the capacity gains and the network topology dynamics that result in scalability and resilience for large-scale deployments when mobile gateways are used for data forwarding from a WSN to remote sink nodes (the command and control server in the scenario targeted in this thesis). The authors consider parasitic mobility such that mobile devices are attached to mobile hosts. This means that the mobility of a device is due to its attachment to a moving host body. This is also the case considered in this thesis. However, the authors only consider controlled mobility patterns, i.e., the mobility is either under the control of the sensor network or the trajectory is pre-configured in a way that enables data collection from the sensor nodes, resulting in limiting the mobility

of mobile hosts. In addition, the authors do not leverage mobile devices to reduce the communication load or energy consumption on the resource-constrained sensor nodes.

In line with the argument presented in this thesis that geographic routing incurs less overhead as compared to the traditional routing where the paths are built by flooding the network with route discovery messages, the protocols discussed in this section rely on an underlying geographic forwarding approach (Ye et al., 2002; Kim et al., 2003).

The protocols presented here deal with only local mobility of the sink with the exception of Kusy et al. (2009), such that the mobile sinks move along the connected edges of the sensor network.

As can be seen in this section, the majority of the protocols try to extend existing routes towards the mobile sink, unless new routes are absolutely necessary. The approach presented in this thesis is inspired by the approach taken by Kusy et al. (2009), where the proxy node of the mobile sink continuously changes with the movement of the mobile sink. Similar approaches will not degrade the performance of the network when the sink is mobile and are capable of providing consistent performance, enabling fast reactivity of the protocol (Table 2.4).

When the data has to be sent to a different network using a mobile gateway, the problem of finding an appropriate gateway among all the available gateways including mobile and stationary gateways, still exists. Using the protocols that cope with sink mobility, the data can be delivered to a mobile gateway, if such a gateway is identified. However, the existing work that deals with sink mobility, places the burden of tracking the mobile devices on resource-constrained sensor nodes, which can result in inefficient performance of the sensor network.

The ROME protocol (Basagni et al., 2010) utilises mobile devices to forward data within the network. However, it restricts the mobile devices to engage in data forwarding activity only when the mobile devices are not moving. This allows the ROME protocol to reduce the communication load on the resource-constrained sensor nodes to some extent (Table 2.4). However, in the envisaged protocol (developed in this thesis), enabling

the mobile devices to forward data within the sensor network is not necessary, as the mobile devices are assumed to be equipped with long-range radio that will enable them to deliver data directly to the command and control server.

None of the protocols discussed in this section aim to reduce communication traffic on congested critical nodes. Ye et al. (2002) somewhat reduce congestion by constructing separate grids for each source, such that each source has its own set of dissemination nodes. In addition, with the exception of Akkaya and Younis (2004), none of the protocols discussed in this section are energy aware either.

Table 2.4 provides a comparison of the protocols discussed in this section and shows that none of the protocols discussed in this section fulfil the requirements (Chapter 1, Section 1.5) of reducing traffic within the WSNs while achieving fault tolerance, energy conservation, congestion reduction and end-to-end delay reduction.

Requirements	TTDD	SEAD	EARM	IAR	Predictive QoS	ROME
Latency Reduction	●	×	●	●	●	●
Device Aware	×	×	×	×	×	○
Energy Conservation	×	×	●	×	×	×
Congestion Reduction	○	×	×	×	×	×
Fault Tolerance	×	×	×	×	×	●
Fast Reactivity	○	○	×	○	●	●
Communication Load Reduction at sensor nodes	×	×	×	×	×	○

Table 2.4: Comparison of routing protocols with mobile devices within the context of WSNs, with envisaged goals and requirements.

Legend: ● = Fully Supported, ○ = Not Fully Supported, × = Not supported.

2.2.5 Hierarchical Protocols

This section explores hierarchical routing protocols due to similarities in their network topology when compared to the hierarchical network topology assumed in this thesis. This section discusses the existing approach taken for the systematic composition of heterogeneous networks.

These protocols leverage heterogeneous networks composed of devices with different sets of communication capabilities, by positioning each category of device in a different logical level of the network hierarchy. By positioning nodes according to their communication capabilities, these protocols form clusters of devices, where a device belonging to higher level becomes a cluster head for lower level devices. With this approach the control overhead is reduced, due to traditional routing approach being taken only within clusters, while inter-cluster routing is performed via cluster heads.

2.2.5.1 Hierarchical Proactive Routing for MANET (HOLSR)

Villasenor-Gonzalez et al. (2005) divide the devices in a network hierarchy based on the number of types of radios that a device is equipped with. For example, devices with a single type of radio belong to level 1, devices with two types of radios belong to levels 1 and 2, devices with three types of radios belong to levels 1, 2 and 3 in the network hierarchy and so on. Devices from the higher levels dynamically assume the role of cluster head for the devices that belong to the lower level(s) of the hierarchy (Figure 2.7). HOLSR is based on the specifications of the OLSR protocol (Adjih et al., 2004), that operates in a flat network topology. The HOLSR protocol demonstrates reduced topological control overhead at different levels of the logical hierarchy as compared to OLSR and a reduction in routing computational overhead. The routing computational overhead is less than OLSR because if a link is broken within a particular cluster, then nodes in that cluster only need to recompute their routing table entries.

A node in the network that belongs to more than one level, i.e., has more than

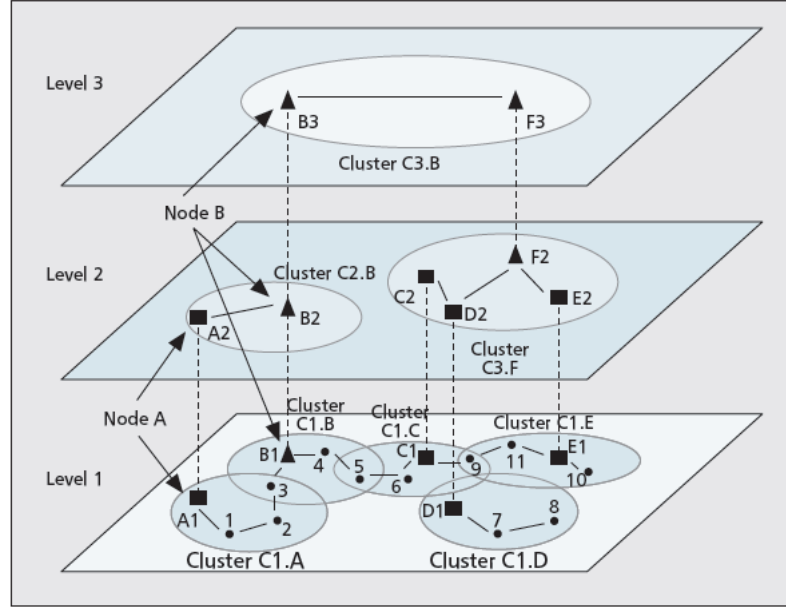


Fig. 2.7: Example of hierarchical network architecture (Villasenor-Gonzalez et al., 2005).

one type of radio, periodically announces its availability to act as a cluster head and invites other nodes to join its cluster by broadcasting a *Cluster ID Announcement* (CIA) message. The neighbouring nodes from lower levels in the hierarchy, upon receiving this message, join the cluster and re-broadcast the CIA message to invite nodes that are further away from the cluster head. If a node receives invitation from more than one cluster, then it joins the cluster which is closest to itself depending on the hop count from the cluster head. A *Hierarchical Topology Control* (HTC) message is used to share cluster membership information with the higher level nodes. This means that the highest nodes in the hierarchy have full knowledge of the network and the number of routing table entries increases proportionally as the obtained network information increases.

In order to perform multi-hop communication within a cluster, the OLSR protocol is employed. Nodes at each level in the hierarchy select *multi-point relays* (MPRs) independently, within the cluster, in order to communicate with nodes within the same

level. A *Topological Control* (TC) message is similarly shared within the cluster itself, independently of other clusters. Based on these TC and HTC messages, routing tables are generated.

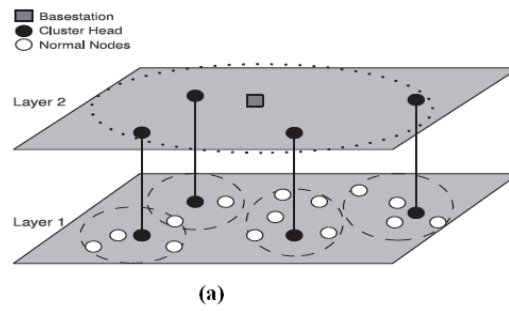
If communication has to be performed within the cluster then the routing tables are employed to forward data along a multi-hop path. If the packet has to be delivered outside the cluster, then the cluster head is used. In the case of mobile nodes, if a node receives a CIA message from a new cluster head that is closer to its current or previous cluster head, then the node joins this new cluster. The HOLSR protocol allows nodes to move around in groups, employing a hierarchical addressing scheme such as IPv4 addresses.

2.2.5.2 Hierarchical Cluster Based Routing

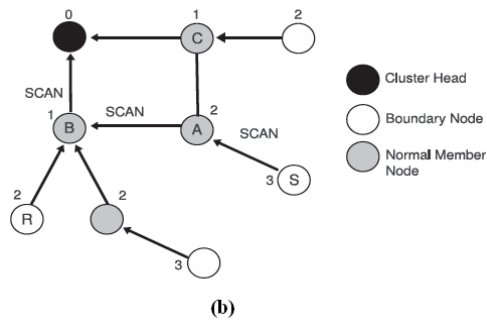
Xia et al. (2009)'s HCB protocol is inspired by HOLSR, however this protocol divides the network in two layers only (Figure 2.8a). The authors empirically demonstrate that HCB has lower control overhead and better delivery ratio than HOLSR in a two-level network.

In this approach, layer 1 contains nodes that have a single short range radio, whereas layer 2 consists of devices with two radios, one short-range and another long-range radio, and are called *super nodes*. The super nodes assume the role of cluster heads for the normal nodes. Each cluster head periodically broadcasts a *Heartbeat* message, containing its address and hop distance to the cluster-head, which is initially set to zero. Normal nodes join the cluster that is closest to them and re-broadcast the heartbeat message after increasing the hop count field. Each node becomes aware of its neighbouring nodes based on these re-broadcast heartbeat messages, along with their distance from the cluster head.

A node becomes a *boundary node* if none of its neighbours in the cluster are farther from the cluster head than itself. The boundary nodes periodically initiate a *scan* message towards the cluster head (Figure 2.8b). The intermediate nodes include themselves



(a) Two layers in HCB.



(b) Boundary nodes and cluster heads in a cluster

Fig. 2.8: Hierarchical Cluster Based Routing (Xia et al., 2009).

in this scan message along with their hop distances from the cluster head. Upon reception of this scan message from the boundary node via intermediary nodes, the cluster head becomes aware of its members and the topology of its cluster. The cluster head then combines the partial information from each of the scan messages that it has received from different boundary nodes, constructs topology information and sends out this topology information in the next heartbeat message.

Each cluster head also exchanges membership information with its neighbouring cluster heads using *Membership* messages, which enables inter-cluster routing. Nodes create routing tables based on the information they received in the *heartbeat* message. If a node cannot find a route from its routing table, then it forwards the packet towards the cluster head. The latter mechanism is employed in particular when the destination is in a different cluster.

2.2.5.3 Summary and General Discussion

The protocols discussed in this section deal with hierarchical structures and device heterogeneity in terms of communication capabilities (Table 2.5), which enables these protocols to reduce information maintenance and route discovery overheads within clusters. Communication within clusters still depends on route discovery or topology maintenance packets initiated by cluster heads in order to let every node know the topology of the cluster. Such topology messages incur large overhead especially in large-scale networks and/or dense deployments, because the size of such messages grow on each hop as each node includes information about itself in the topology control message. A large number of topology control messages may be generated in these protocols in a highly dynamic environment where new mobile devices may join or existing mobile devices may leave the network frequently.

Although these protocols inherently reduce the information maintenance requirements on the lower-level nodes by reducing the size of the routing tables, they do not reduce the traffic load on less capable devices. The traffic is only routed towards the

cluster heads or more capable devices when the data has to be sent to a device that belongs to a different cluster (Table 2.5).

Also, these protocols only differentiate between different types of devices based solely on their communication capabilities (number of radios) and do not account for their energy reserves or mobility characteristics e.g., vehicular or walking speed, or stationary devices.

Table 2.5 summarizes that the hierarchical routing protocols, discussed in this section, do not consider fault tolerance and fast reactivity as their top priority, due to their reliance of route discovery procedures. HCB can potentially react faster to the changing environment, because the topology information is shared in the periodic heart beat (beacon) messages, and any changes to the network topology may be detected quicker than in HOLSR protocol.

Requirements	HOLSR	HCB
Latency Reduction	●	●
Device Aware	○	○
Energy Conservation	×	×
Congestion Reduction	×	×
Fault Tolerance	×	×
Fast Reactivity	×	○
Communication Load Reduction at sensor nodes	×	×

Table 2.5: Comparison of hierarchical routing protocols with the envisaged goals and requirements.

Legend: ● = Fully Supported, ○ = Not Fully Supported, × = Not supported.

2.2.6 Delay-tolerant Protocols

Delay tolerant routing protocols show resemblance to the envisaged goal of delivering data packets by using mobile devices that come in contact with the sensor nodes.

This class of routing protocols are typically used to deliver data in intermittently connected networks, comprised of mostly mobile nodes. In such networks, it is not expected that a path between source and destination necessarily exists. However, there might be a possibility to forward data through nodes that come into contact with each other at some point. In such networks, the nodes store packets until they come into contact with nodes that have a higher probability of delivering data to the destination, and transfer packets to those nodes.

2.2.6.1 Epidemic routing

Vahdat and Becker (2000) present a routing protocol that relies on the theory of epidemic algorithms, for such networks. In such networks, the data packets spread in the network like an epidemic, eventually reaching the destination.

The messages are stored at nodes and a summary of the messages stored is kept in the *summary vector*. When a node comes into contact with another node, these summary vectors are exchanged. Each node can then determine which messages it has not previously seen and can receive such messages. Each message is identified by a unique *global ID*. A time to live (TTL) is also maintained for each message, which allows the spread of messages throughout the network to be controlled at design time. If this TTL is large enough, a message can spread throughout the entire network, eventually reaching the destination, but will incur high memory usage.

2.2.6.2 PRoPHET

Lindgren et al. (2004) propose a probabilistic delay-tolerant routing approach, PRoPHET, based on the argument that the mobility patterns of users are not always random, and

could be predictable. For example, if a user has visited a location several times before, then the user may be likely to visit that particular location again. The authors use this abstraction to limit the memory and bandwidth usage, due to which epidemic routing (Vahdat and Becker, 2000) is not the ideal choice.

In order to leverage mobility patterns, they use a metric called *delivery predictability*, maintained at every node a , for each known destination b . This metric is the probability of an encounter of a with b or the probability that a will be able to deliver data to b , and increases with each encounter of a and b . The protocol also exploits transitive nature of encounters, i.e., exploiting indirect contacts. Old encounters are degraded over time.

At each encounter the nodes exchange a *delivery predictability vector*, in addition to a *summary vector*, which contains information about known destinations. A message for a particular destination is forwarded to a node with a higher delivery predictability. A forwarded message is not removed, resulting in multiple copies of a message being present in the network. The authors evaluate their approach against epidemic routing (Vahdat and Becker, 2000) and show that their approach leads to fewer message exchanges and lower communication overhead.

2.2.6.3 Social Network Routing (SimBet)

Daly and Haahr (2007) propose social network analysis techniques to store, carry, and forward packets for such intermittently connected networks. The authors do not assume knowledge about the whole network topology while routing packets in the network and only use locally available information on each node. In particular, the authors exploit '*Betweenness Centrality*' and '*Similarity*' in order to exchange messages between encountering nodes. *Betweenness Centrality* measures how much control a node has over the information flowing between others, representing the nodes that a particular node comes in contact with. *Similarity* represents the common neighbours that two nodes share. Based on these two metrics, the protocol calculates the *SimBetUtility* for encountering nodes for particular destinations. A node forwards a message to an encountered

node that has better *SimBetUtility* for the message's destination. This protocol keeps only one copy of the message in the network.

The authors evaluate their approach against epidemic routing (Vahdat and Becker, 2000) and P_{Ro}PHET (Lindgren et al., 2004), and show that their approach is generally better than P_{Ro}PHET in terms of message delivery, end-to-end delay and number of hops per message, however comparable to epidemic routing incurring 40% more delay. The protocol performs best in terms of total number of forwards as compared to epidemic and P_{Ro}PHET routing protocols.

2.2.6.4 Sensor Context Aware Routing (SCAR)

Pasztor et al. (2007) propose a delay-tolerant energy-aware routing protocol for the sensor networks. The authors predict the delivery probability by using Kalman Filter based on the changing connections with neighbouring sensors, co-location with the sink nodes, available energy reserves and available buffer slots. In this protocol the bundles are forwarded to multiple nodes, such that the node that has the highest probability of delivery retains the master copy, whereas, other nodes mark their copies of the packet as backup copies. Upon coming into contact with another node, the node sends a hash table containing the identifiers of the packets. These hash tables are similar to summary vectors in nature as discussed in previous protocols. If the buffer of a node is full with master copies of packets, then the node does not advertise its delivery probability. However, the node accepts packets from its neighbours if it has empty slots or backup copies, as the backup copies can be replaced by master copies of other packets. In this protocol, the packets are exchanged among neighbouring nodes if the delivery probability of the next hop node is much larger than the previous hop node.

The authors compared their technique to a random replication based approach, where the packets are replicated to randomly selected neighbours and offloaded to the sink when the sink is in range. The evaluation of the protocol shows better performance of the SCAR protocol when compared to random replication technique.

2.2.6.5 Summary and Discussion

Each node in these protocols maintains information about every other node, that it directly or indirectly comes in to contact with. If such protocols are used in an established and connected network, then every node in the network will hold information about every other node. This will typically result in high memory requirements on each node, which may not be feasible for WSN nodes that have very limited memory and computational resources (Table 2.6). For this reason, these protocols are especially suitable where nodes rarely come into contact with each other.

Requirements	Epidemic	PRoPHET	SimBet	SCAR
Latency Reduction	×	●	○	○
Device Aware	×	×	×	×
Energy Conservation	×	×	×	●
Congestion Reduction	×	×	×	○
Fault Tolerance	●	●	●	●
Fast Reactivity	●	●	●	●
Communication Load Reduction at sensor nodes	×	×	×	×

Table 2.6: Comparison of delay-tolerant routing protocols with the envisaged goals and requirements.

Legend: ● = Fully Supported, ○ = Not Fully Supported, × = Not supported.

In addition, since *summary vectors* or similar information vectors are exchanged when the nodes come into contact, in a connected network, these *summary vectors* will be shared continuously. If the *summary vector* sharing is optimized to occur either periodically or when something changes in the connected network, even then due to the large number of devices in the connected network, the size of such *summary vectors* could be large. This could potentially result in fragmentation of packets, especially on radio links with small MTU (maximum transmission unit), which in turn results in more

packet transmission. Since energy reserves on each node decrease with every transmission and reception, such protocols could result in depletion of the energy at sensor nodes due to transfer of summary vectors.

However suitable these protocols are to provide fault tolerance, they do not address reduction in communication load on resource-constrained sensor nodes, as typically these protocols are not required to differentiate among different types of devices present in the network. As the network is intermittently connected, congestion is not expected and energy conservation is not a priority in such scenarios. An exception to this general observation is the SCAR (Pasztor et al., 2007) protocol that includes energy reserves in its decision metric to perform energy efficient routing even in the intermittently connected network.

These protocols address a very interesting problem of data delivery under environments where proper network coverage is not available. These protocols generally use the technique of locally storing and forwarding data packets when connectivity towards the destination becomes available. Such approaches could be used to extend the STERIOD protocol presented in this thesis, such that the sensor nodes may store data packets until a more capable mobile device becomes available or the storage space on the sensor node is fully consumed or until a certain configured “time to stay” for a data packet expires.

Table 2.6 summarizes the properties of delay tolerant routing protocols, discussed above, against the requirements set out in Chapter 1, Section 1.5 that are necessary to obtain a solution that can offload communication load away from resource-constrained sensor nodes. Table 2.6 shows that PROPHET and SimBet routing protocols reduce latency by using the most suitable forwarding candidates, whereas SCAR uses energy-aware metric when using Kalman Filter based techniques for decision making.

2.2.7 Device Aware Protocols

The routing protocols in this category are aware of the capabilities of devices. The main aim of these protocols is to reduce data transmissions through less capable nodes.

In such protocols, the packets are redirected towards more capable devices, which are assumed to have longer transmission range and be more reliable.

The protocols discussed in this section, use traditional routing approaches, such as AODV (Perkins and Royer, 1999) and DSDV (Perkins and Bhagwat, 1994), as their underlying routing mechanism. These protocols assume that the more capable nodes can increase their transmission range to reach a wider geographic area and use asymmetric links between nodes having different communication capabilities. Typically, these protocols differ in the way traffic is redirected towards the more capable nodes in the network.

2.2.7.1 DEAR

Avudainayagam et al. (2003) present an energy-aware routing approach that differentiates between battery-powered and line-powered devices. The authors build on shortest path energy-aware routing protocols, e.g., a variant of AODV (Perkins and Royer, 1999) that favours relay with better energy reserves, if multiple relays to the same destination with the same hop count are available, and redirect traffic to a line-powered device if the discovered delivery cost to the line-powered device is less than the discovered delivery cost to the intended destination. Apart from a routing table, the authors also maintain a redirect table, which maps destination addresses to the addresses of line-powered devices. The redirect table is updated at the same time as the routing table, where the routing table distinguishes between line-powered and battery-powered devices. When a packet has to be sent, the routing table is consulted to check if the packet could be redirected to a line-powered device.

The authors assume that a line-powered device can increase its transmission power to reach all the devices in the network in a single hop. For a given packet, if redirection is available, then the packet is routed to the line-powered device indicated in the redirection table for the destination of the packet. The line-powered device, on reception of the packet, increases its power and sends the packet to the destination in a single hop. If

redirection for a given packet is not available, then the packet is routed towards the intended destination using an energy-aware protocol.

Discussion: By redirecting the packets toward line-powered devices, this protocol tries to reduce the forwarding load on devices with limited energy reserves. However, the nodes that are located near line-powered devices could potentially suffer due to this scheme, because many packets could be redirected towards line-powered devices. This may also increase congestion around line-powered devices, with the increase in traffic load.

DEAR differentiates devices into two categories only, which does not provide the required flexibility in the scenario of this thesis, because of the presence of mobile devices that are battery-powered but are also rechargeable and do not fit in either category. As the more capable devices are assumed to be line-powered, the protocol does not consider mobility of the more capable devices. For this reason, the DEAR protocol is unable to take advantage of mobile devices to offload communication load on resource-constrained sensor nodes.

Because DEAR builds on shortest path protocols, it incurs route discovery overhead. If the environment is highly dynamic (as is the case in this thesis) and devices appear and disappear frequently, the control overhead could increase exponentially, as also discussed in Section 2.2.1. In addition, line-powered devices are assumed to be able to cover the entire network, when their transmission power is increased. However, this may not be the case in deployments covering large geographic areas.

2.2.7.2 MC-Routing

Du et al. (2006) present a device aware multi-class routing protocol for mobile ad hoc networks in which the more capable nodes are also mobile, unlike the DEAR protocol discussed above. The protocol differentiates the devices into two categories of general nodes and backbone-capable nodes, based on their transmission range, data rate, pro-

cessing capability and reliability. The backbone-capable nodes are assumed to be better in all aspects than the general nodes (G-nodes). The authors assume a fixed coverage area that is divided into cells of fixed dimensions, where all cells are equal. Each cell in the coverage area can have both type of nodes, but only one single backbone-capable node is elected as the backbone node (B-node) for a given cell. All the nodes are assumed to be mobile, but they are assumed to remain within the coverage area. A backbone node is assumed to communicate with other backbone nodes directly in adjacent cells.

The Election of a B-node in each cell is triggered either by a B-node that is leaving the cell or by a G-node that notices that no B-node is present in the cell. The election message is flooded to all the nodes in the cell. When a backbone-capable node receives an election message, it broadcasts a claim message after a random timer expires. A backbone-capable node does not broadcast its claim message, if it receives a claim message from another backbone-capable node. The random timer is used to reduce concurrent broadcasts of claim messages from distinct backbone-capable nodes in the cell. All nodes in the cell are assumed to receive the broadcasted claim message. After a successful election of a B-node, all other backbone capable nodes are assumed to act as a G-node, using reduced transmission range (power).

Every node shares its location with the nearest B-node. The B-node then broadcasts the location packet within a small hop count. All the B-nodes also share their information periodically with a special B-node (B_0). The B_0 node is assumed to be either stationary or have very limited mobility. The B-node in the source cell requests the location of the destination from B_0 using the routing protocol discussed below.

If the source node is a G-node, it floods the cell with a route request packet. When the route request reaches the B-node within the cell, the B-node transmits a route reply packet directly to the G-node, containing the path to the B-node.

The cell in which the destination node is present is identified using a location sharing service. The B-node then draws a straight line towards the destination cell from its own cell. The B-node then sends a route request to the B-nodes in adjacent cells that fall

under the drawn line. The route request is forwarded towards the destination cell, by the intermediary B-nodes. When the route request reaches the B-node in the destination cell, it sends back a route reply that includes the path to the destination cell. If the density of backbone capable nodes is small, i.e., not all cells may have a B-node, then the source B-node multicasts the route request via adjacent cells that do not fall under the straight line to the destination cell.

When data reaches the B-node in the destination cell, it will simply transmit the data packet directly to the destination node, as its transmission range is assumed to cover the dimensions of the cell. The destination node (if it is a G-node) will flood the cell with ACK packet that will reach the B-node, because G-node has lower transmission range than the B-node.

Discussion: The protocol redirects the packets towards more capable backbone nodes (B-nodes) and reduces data forwarding by less capable G-nodes in adjacent cells. Because long range communication is mostly carried out by B-nodes, the protocol also reduces end-to-end delay and routing overhead in addition to improving throughput.

However, the G-nodes that are present near the B-nodes may experience congestion due to traffic arriving towards the B-node of the cell. As the election algorithm only allows a single B-node per cell, the congestion experienced by G-nodes near the B-node cannot be alleviated, even in the presence of other backbone-capable nodes.

Using backbone-capable nodes, network partitions can be mitigated with this protocol. However, due to a reliance on route discovery and route repair algorithms, this protocol cannot quickly adapt to the changes in the network in highly-dynamic environments. In addition, due to flooding during route discovery and route repair, the protocol may incur undue control overhead, especially in highly-dynamic environments.

Unfortunately, the MC-Routing protocol does not take energy consumption of network nodes in to account. This discrepancy can result in less capable nodes getting depleted quickly. In addition, a new B-node will not be used if the old B-node gets

depleted, as mobile devices also have limited (though rechargeable) energy reserves.

The authors also introduced a new media access control (MAC) protocol that divides the frame into three slots where transmissions are scheduled in each slot according to device types. Because no inactive periods are scheduled in the frame, all devices will always listen, reducing the energy efficiency of the protocol. In addition, all nodes are assumed to remain in the coverage area, meaning that the protocol does not consider appearance and disappearance of nodes.

For these reasons, this protocol is not particularly suitable for sensor networks, where the nodes are very resource constrained with very limited energy reserves.

2.2.7.3 DELAR

Inspired by the Multi Class Routing protocol (Du et al., 2006), Liu et al. (2011) presented the DELAR protocol that differentiates the devices in two categories, P-nodes and B-nodes, where P-nodes are assumed to be more capable and powerful nodes, whereas B-nodes are assumed to be base nodes. The authors base the protocol on the pro-active routing protocol DSDV (Perkins and Bhagwat, 1994) that applies shortest path routing. The route discovery and route repair mechanisms are similar to the traditional routing protocols.

The DELAR protocol uses energy and load aware metric for packet routing and next hop selection. The protocol considers the remaining energy on the device, the number of packets in the transmission queue and the energy spent per packet, in order to achieve energy efficient routes.

The more capable P-nodes are assumed to have much better energy reserves and communication capabilities in this protocol as well. For this reason, the packets are routed towards the P-nodes which can deliver the packet to the destination in less hops, unless the P-node is further than the destination.

The DELAR protocol also uses asymmetric MAC protocol, as the one used in MC-Routing protocol, due to asymmetric links between P-nodes and B-nodes. This protocol

introduces a concept of backward paths which are mostly used for delivering acknowledgements (ACKs) and CTS packets towards a P-node from a B-node.

As the P-nodes are assumed to have better communication capabilities than B-nodes, the protocol defines three transmission ranges as follows:

1. From B-node to B-node transmission range, given by T_{bb} .
2. From P-node to B-node transmission range, given by $T_{pb} = n \times T_{bb}$.
3. From P-node to P-node transmission range, given by $T_{pp} = m \times T_{bb}$.

In the above mentioned transmission ranges, it is assumed that $T_{pp} > T_{pb} > T_{bb}$ as $m > n$. The transmission range from a B-node to a P-node is the same as T_{bb} . In this way, the protocol reduces the hops and energy consumption on B-nodes, once a packet arrives at a P-node that in turn can boost its transmission range and send packet directly to the destination or another intermediary P-node.

In order to reduce collisions, the protocol divides a single superframe into three portions of time, i.e., P-to-P, P-to-B, and B-to-B communication portions. The P-to-B communication portion of the superframe is variable and depends on the number of P-nodes available in the network. As the number of P-nodes increase, this portion of the superframe also increases. During the network formation, the P-nodes will use higher transmission range/power to negotiate slots in the superframe among themselves and then will broadcast these results to the resource-constrained B-nodes. The authors assume perfect time network synchronization in order to allow communication within allocated portions of the superframe.

As the B-nodes are assumed to be resource-constrained devices, the authors turn-off the radio at B-nodes during P-to-P portion of the superframe, in order to conserve energy.

Discussion and Shortcomings: The DELAR protocol is mainly designed for low mobility ad-hoc networks, i.e., the connections do not break often, however, all nodes

are considered to be mobile.

The protocol does not include any inactive portion in the superframe, in order to reduce the energy consumption due to the listening task performed on the devices. Although the protocol does allow the B-nodes to turn off their radio during the P-to-P portion of the superframe, the energy consumption is still expected to be high as the P-to-P portion is smaller than the remaining superframe, resulting in a small interval during which the radio of the B-nodes is switched off.

Increasing the number of P-nodes in the network increases the size of the superframe as more P-to-B slots are required to be allocated to individual P-nodes. This results in an increase in the end-to-end delay, which is also reflected in the evaluation of this protocol.

The protocol attempts to reduce congestion by employing the queue length in the routing metric. This means that a device which has more packets in the queue at a given time will not be the preferred next hop. This metric will not, however, take the load that a device may experience into account, and therefore cannot support pro-active load reduction on congested nodes.

The backward routing paths introduced in DELAR protocol allow the ACKs and CTS packets to reach the source in unidirectional links, however, this also introduces the possibility of stale CTS signals, such that by the time a CTS is received at a P-node from a B-node, the radio channel may not be free anymore.

Due to the reliance on route discovery and repair procedures that are based on the AODV and DSDV protocols, the routing and discovery overhead could increase with the increase in mobility. Possibly, for this reason, Liu et al. (2011) report higher end-to-end latency and packet losses as the mobility increases, which is an acceptable trade-off.

The DELAR protocol is used as the baseline to evaluate the protocol proposed in this work because it is the most recent work that redirects the traffic towards more capable devices.

2.2.7.4 Summary and General Discussion

Table 2.7 summarizes the characteristics of the device-aware routing protocols for the requirements set out in Chapter 1, Section 1.5 that are necessary to obtain a solution that can offload communication load away from resource-constrained sensor nodes.

Table 2.7 shows that all the device aware protocols reduces end-to-end latency by using traditional routing protocols when the number of hops to the destination are reduced. MC-Routing does not perform energy-aware routing and only DELAR routing protocol somewhat considers the load on the resource-constrained devices.

Table 2.7 also shows that even though these device-aware routing protocols reduce communication load on resource-constrained node by redirecting traffic towards more capable nodes, the control overhead can be huge in highly dynamic environments due to their tendency to flood the network with routing related control messages.

All three protocols discussed in this section are not considered to be completely fault tolerant, i.e., they cannot quickly build alternate routes in case of network partitions, due to their reliance on route discovery and route repair procedures, even in the presence of more capable nodes that can be alternatively used.

2.2.8 Other

This section discusses a few communication protocols that are generally interesting to study and provide alternative perspective to typical communication paradigm.

An interesting protocol is discussed in this section that takes advantage of the inherent broadcast nature of wireless communication. In such protocols, only the nodes that are capable for forwarding the packet towards the destination may contribute to the data delivery, whereas, in case of communication failure the source node does not have to retry packet transmissions in order to reach the destination.

This section also discusses asynchronous MAC layer protocols, with particular focus on X-MAC (Buettner et al., 2006), that allow the network nodes to communicate with

Requirements	DEAR	MC-Routing	DELAR
Latency Reduction	•	•	•
Device Aware	•	•	•
Energy Conservation	•	×	•
Congestion Reduction	×	×	○
Fault Tolerance	×	○	○
Fast Reactivity	×	×	×
Communication Load Reduction at sensor nodes	○	○	○

Table 2.7: Comparison of device-aware routing protocols with the envisaged goals and requirements.

Legend: • = Fully Supported, ○ = Not Fully Supported, × = Not supported.

each other without the need to synchronize their transmission and reception schedules.

2.2.8.1 ExOR

Biswas and Morris (2005) presented a routing protocol that defers the choice of each hop until after the transmission for that hop. This means that the source node broadcasts a packet towards the destination and the receiving node, which is closest to the destination (and has received the packet successfully), will forward it onward. In this way, the next-hop is only selected after the packet is received successfully by multiple nodes.

This protocol takes advantage of dynamism in the network such that if the closest node to the destination has not received the packet, then the second closest node can forward the packet. Conversely, if a node closer to the destination becomes available, which was not present earlier and was not known to the rest of the network, it can forward the packet instead of the node that was known by the network but is not as close to the destination as this newly found node.

The receiving nodes must agree among themselves on which subset of the nodes

actually received the packet successfully and then which node is to be responsible for forwarding a particular packet, so that only one node forwards the message in order to avoid congestion and contention. Since this agreement involves communication, which the authors acknowledge as an overhead, a mechanism within the protocol is required to minimise the communication and reach an agreement at the same time. The authors address this overhead by including a list of candidate forwarders in the packet from the source node, prioritised by their estimated cost to the destination. In this way, the node that has the highest priority in the list of candidate forwarders, will re-broadcast the packet towards the destination and include its own list of candidate forwarders that could further advance the packet.

All the forwarding devices listed in the candidate forwarders' list listen for acknowledgements by the higher priority nodes for each packet and forward the packets that were not acknowledged by higher priority nodes, by rebroadcasting those packets. In this protocol, a source transmits a batch of packets towards the destination in order to reduce the number of transmissions.

Discussion: This mechanism of having a list of candidate forwarders, batching packets in the message and listening for acknowledgements by higher priority forwarders, implies that the protocol uses large packet sizes and probably utilises promiscuous mode, as the receiving nodes would have to keep their radios in listening mode in order to receive broadcast transmission.

Large packet sizes and promiscuous mode when used either individually or when combined together could possibly result in higher energy consumption. Chipcon (2006) shows that during listening mode, the CC2420 radio consumes more current than during transmission mode. CC2420 (Chipcon, 2006) is a low-power radio module that is widely used in a number of sensor platforms such as (Crossbow, 2005; SunSpot, 2004).

Thus, this approach of broadcasting packets instead of unicasting towards a particular forwarder, may not be suitable for resource-constrained WSN devices.

Requirements	ExOR
Latency Reduction	●
Device Aware	×
Energy Conservation	×
Congestion Reduction	○
Fault Tolerance	●
Fast Reactivity	●
Communication Load Reduction at sensor nodes	×

Table 2.8: Comparison of ExOR routing protocol with the envisaged goals and requirements.

Legend: ● = Fully Supported, ○ = Not Fully Supported, × = Not supported.

As shown in Table 2.8, the protocol does not take device characteristics in to account when forwarding towards the destination, and only tries to advance packets through the closest possible node. This means that a device that is more capable may not be utilized if it is not the closest one to the destination. Thus, this protocol as well, does not aim to reduce communication load on less capable nodes.

Table 2.8 shows that the ExOR protocol is not energy aware and does not reduce communication on resource-constrained sensor nodes. However, because the ExOR protocol uses broadcast based communication model, it can react fast in highly dynamic environments in addition to providing fault tolerance in case of network partitions by using devices that may appear in the vicinity of the network without depending on route discovery procedures.

2.2.8.2 Asynchronous MAC Layer

The asynchronous MAC protocols (Buettner et al., 2006; Wang et al., 2007) provides an alternative to the synchronous MAC protocols such as IEEE802.15.4 (2006). The

asynchronous MAC protocols allows the nodes in the network to communicate with each other without the need for them to synchronize their transmission and reception schedules. Typically, the receiving nodes follow a low power listening schedule such that these nodes duty cycle their listening periods. The node that wishes to send data to a certain node, transmits a long pre-amble that is a bit longer than the listening duty cycle. Upon the completion of the pre-amble the sending node transmits the data packet. The receiving node keeps its radio awake after receiving the pre-amble to receive the data packet.

Buettner et al. (2006) introduced X-MAC, a short pre-amble approach to asynchronous MAC protocols. The authors highlight that in typical asynchronous MAC protocols, the nodes suffer from over hearing problem such that all nodes remain awake after they receive the long-preamble even if the data packet is not targeted at them. Additionally, if multiple nodes wish to transmit data packets, then all such nodes will transmit their pre-ambls, which could potentially result in keeping the receiving nodes awake all the time as they would expect a series of packets in a dense network deployment. For this reason, Buettner et al. (2006) include the target node's address in their short pre-amble that is repeated periodically, until the transmitting node receives an acknowledgement from the target node. The receiving node keeps its radio in the listening mode a little longer after the reception of the data packet, so that if another transmitter wishes to send data to the same node, it can easily receive data packet from the other node as well without the need to wait for a separate pre-amble.

Similar techniques are also used by Wang et al. (2007) where they use IDLE radio mode to conserve energy as its transition from transmission and reception mode is quite fast.

Discussion: Though both alternatives of synchronous and asynchronous MAC protocols have their advantages and disadvantages (few discussed below), the choice of the MAC layer depends on multiple factors such as environment, application, available hard-

ware etc.

For example, if the available hardware consumes less energy in listening than in transmission then asynchronous MAC layer can be adopted such that the sensor nodes can duty cycle their listening periods, whereas, the source nodes only need to transmit pre-amble when they have some data to send. Similarly, asynchronous MAC protocols can be ideal in situations where the network topology changes often such that no network structure can be formed, e.g., in MESH network topology.

The synchronous MAC protocol can be used when the energy consumed during transmission can be configured to be less than the energy consumption in reception such that periodic beacons can be broadcasted without high energy penalties. In environments, such as the one discussed in this thesis, where network structure can be formed, synchronous MAC protocols can be advantageous such that synchronization only needs to occur during the initial network formation phase or when an existing link breaks in order to find alternative paths.

The dynamism in this thesis is introduced due to frequent appearance and disappearance of mobile devices, however, the sensor network is assumed to demonstrate an underlying network structure (cluster-tree topology) even if individual links are not assumed to be stable. In order to take advantage of the mobile devices and divert traffic towards more capable mobile devices, the sensor network needs to discover the said mobile devices before utilizing them. The Inverse Location-Aware Discovery mechanism discussed in Chapter 3, Section 3.3.4 uses synchronous MAC protocol such that sensor nodes periodically broadcast beacon packets to allow mobile devices to discover them easily, such that the transmission power at the sensor nodes is configured at the minimum possible value for the available hardware to conserve energy.

Another reason to use IEEE802.15.4 (2006) synchronous MAC layer in this thesis was its standardization such that the same work can be ported to different hardware platform, if required. In addition, to author's experience, synchronous MAC layer allows the energy consumption due to control overhead to be more deterministic as the beacon interval

and active listening periods are configured at the network design time as discussed in Section 3.3 and Section 4.1 in Chapters 3 and 4 respectively.

2.2.9 Summary

As the WSNs are resource-constrained networks formed by devices that are very low on energy budget and have very limited processing and communicating capabilities, protocols are required that reduce the load on such networks. The existing communication protocols, in particular routing protocols, try to reduce energy consumption by using duty cycling and energy-aware decision metrics, in addition to reducing latency by making smart decisions. Even though potent, unfortunately these techniques are not sufficient to reduce the communication and processing load on such resource-constrained devices, because even with load balancing, the resource-constrained sensor nodes have to process and forward data packets towards the destination in multi hop networks. In addition, intelligent decision making also requires statistics to be stored on such resource-constrained devices, in turn increasing memory requirements.

Similarly, very few protocols are actually capable of coping with network partitions opportunistically and react to changing environments quickly. The majority of the routing protocols capable of efficiently bridging network partitions fall under either the geographic or delay-tolerant category. However, they lack in other aspects such as being device or energy aware, or being capable of reducing congestion at critical nodes.

For these reasons, this thesis suggests utilizing more capable mobile devices that are evermore widely available in modern urban environments, to reduce communication within WSNs. These mobile devices can be used to bypass a number of hops and deliver sensor data directly to the destination using long-range radios. By doing so, a reduction in latency, in energy consumption at resource-constrained sensor nodes, in congestion at critical nodes, in communication within WSN, and provision of fault tolerance in case of network partition can be achieved.

As can be seen from the discussion in previous sections, none of the existing protocols

fulfil the requirements set out in Chapter 1, Section 1.5 that are necessary to obtain a solution that can offload communication load away from resource-constrained sensor nodes.

Even the systems with mobile devices, only target the mobile sink scenario, where the sensor data has to be delivered to the mobile sink. However, in these systems, resource-constrained sensor nodes are responsible for delivering the data to the mobile sinks. In this case, nodes near the mobile sinks may deplete their energy reserves. These protocols do not reduce congestion at critical sensor nodes. Apart from Basagni et al. (2010), protocols with mobile devices do not cope with network partitions either.

Even though device-aware protocols (Avudainayagam et al., 2003; Du et al., 2006; Liu et al., 2011) reduce communication on less capable devices, they are not suitable for WSNs due to their limitations on energy conservation and congestion reduction. In addition, these protocols are unable to react quickly to the changing environment while providing consistent delivery ratio, due to their reliance on route discovery procedures.

Thus, this thesis explores the possibilities of taking advantage of mobile devices to reduce traffic on less capable sensor networks in environment monitoring applications within the context of smart cities. While achieving this objective, this thesis also considers the placement of protocols and mechanisms in order to reduce processing and memory requirements on the sensor nodes. Additionally, algorithms and mechanisms developed in this thesis make conscious choices in achieving distinct objectives of fault tolerance, energy conservation, congestion reduction and latency improvement, depending on the network conditions.

Due to unsuitability of traditional routing approaches that depend on either flooding the network with route requests or utilize geographic forwarding, this thesis takes its inspiration from Kusy et al. (2009) that builds information potentials towards the destination. Similar logical constructs can quickly adopt in highly dynamic environments in a few iterations. In addition, energy and device aware metrics can also be incorporated in these logical constructs that can essentially show the direction that a packet may

take to reach the destination efficiently while reducing the load on resource-constrained sensor nodes.

Arguably, the envisaged routing protocol not only has to take energy aware decisions while routing the packets, but the design of the routing protocol must also take energy conservation into account. This means that, wherever possible, the tasks that consume energy must be placed at more capable mobile devices.

Chapter 3

Design

This chapter first discusses the overall approach taken and the challenges associated with designing a routing protocol capable of reducing the communication overhead from the resource-constrained sensor nodes and shifting this load towards more capable mobile devices in Section 3.1. The problem is decomposed into four major components required to achieve such a routing protocol.

Section 3.2 discusses the network structure and network formation phase. It discusses the overall architecture of the WSN and the process of sensor node joining the network.

Section 3.3 begins by introducing the discovery mechanism available in IEEE802.15.4 (2006) standard. A detailed analysis of discovery mechanisms where the discovery is carried out by sensor nodes and later by mobile devices, is carried out next. Section 3.3 finally discusses the design of the “Inverse Location Aware Discovery” mechanism that places the burden of discovery on the more capable mobile devices, essentially covering the “neighbourhood discovery” component of the overall design space, while keeping the operations at the sensor node energy efficient.

Section 3.4 discusses the design of the “Smart Traffic Energy and Resource aware Offloading using Inverse Discover” routing protocol in detail. It discusses the performance indicators associated with each device and credential update mechanisms in Sec-

tions 3.4.1 and 3.4.2, covering the “Performance indicators” and “Credential and risk feedback” components of the overall design. Section 3.4.3, then discusses the decision making component and how this component is divided amongst the resource-constrained sensor nodes and more capable devices. The parent and route selection is performed on the sensor nodes as discussed under Section 3.4.4, whereas, the mobile device selects a sensor node to achieve individual objectives as discussed under Section 3.4.5. Since decision making is the most important component, Sections 3.4.4 and 3.4.5 also discuss which decisions are taken during which phases of the network life cycle. Section 3.4.5 also discusses how different policies are invoked in terms of strategies depending on the network conditions by introducing prioritized objectives, explicitly focusing on conciliating and achieving the requirements set out in Chapter 1, Section 1.5.

Finally, Section 3.5 summarizes this chapter and shows the memory consumption on resource-constrained sensor nodes when the protocols discussed in this chapter were implemented on TelosB (Crossbow, 2005) sensor network platform for verification.

3.1 Design Overview

Solving the problem described in Chapter 1 essentially requires answering the question of how the forwarding capability of a newly-arrived device can be judged and compared to that of existing neighbours, in a highly dynamic environment. Keeping in mind the unstable nature of the prospective routes, the protocol cannot rely on explicit route discovery procedures (Section 2.2). Similarly, a pure geographic routing approach is not suitable as more capable devices may not be the closest ones to the destination. Hence, the routing protocol should be able to react fast to the changing environment where new and previously known nodes can leave and frequently join the network and change the network topology. In addition, the routing protocol should be able to identify and favour more capable devices.

The scale of the problem considered here, as well as the number of parameters in

consideration, i.e., the energy level of different nodes, the mobility characteristics, the communication, storage and processing capabilities, the trajectory of mobile devices, the data delivery ratio etc, mean that it is typically intractable to create a model of the environment. Indeed, the effect of selecting a particular next hop cannot be fully predicted as all operating conditions cannot be predefined and hence the correct behaviour cannot be decided at design time. Reinforcement learning (RL) (Sutton and Barto, 1998) is considered particularly suitable for such problems (Tesauro, 2007), and therefore its use in this context was explored. Learning, however, is not very suitable for the problem addressed in this thesis due to the energy cost associated with the learning (exploration) phase and the frequent state changes due to the appearance and disappearance of devices. Hence, the solution discussed in this thesis is only partially inspired by RL, particularly in the way the feedback loop is used to judge the previous performance of devices and the four different prioritized policies are invoked depending on the network condition to outsource data forwarding via more capable mobile devices while reducing the communication load on resource-constrained sensor nodes. Reinforcement learning has been previously used in the packet routing domain (Dowling et al., 2005; Forstert and Murphy, 2008; Boyan and Littman, 1994), however some notion of reaching a stable state is assumed, contrary to this research where the stable state remains elusive, due to the mobility of more capable mobile devices and their appearance and disappearance due to the users' control over the communication capabilities of such third-party devices.

The problem is decomposed into four major components (Figure 3.1). The arrows in Figure 3.1 show dependencies and influences among different components. Each of the components is discussed below along with some of the design challenges that they incur.

3.1.1 Decision Making

This component is concerned with the next hop selection, enabling competition among potential forwarders, corresponding to the action selection in RL. Depending on the action selection policy, this component can occasionally allow the current-hop (node

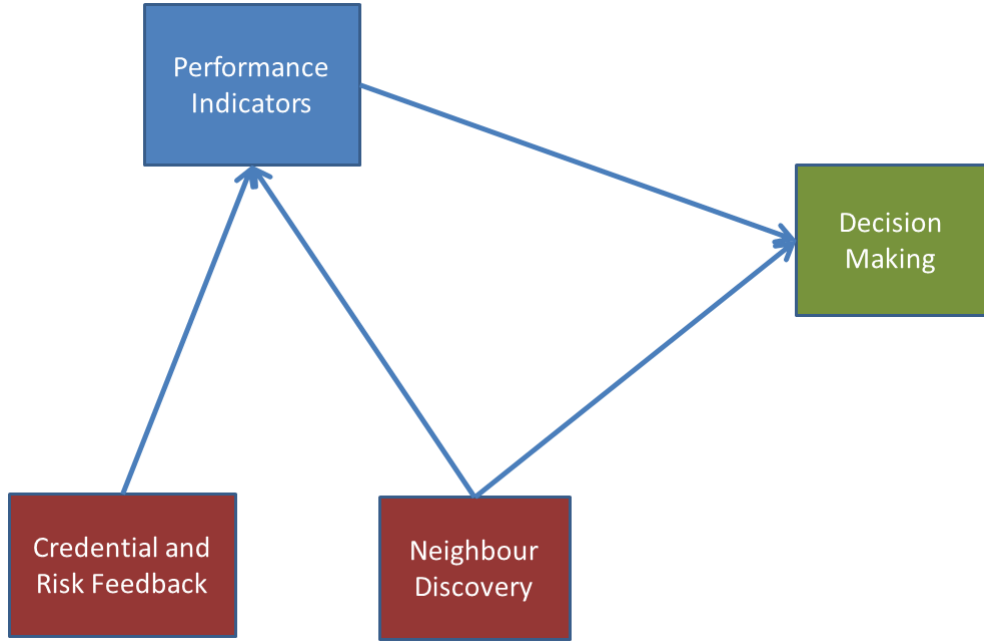


Fig. 3.1: Solution components.

holding the packet) to explore sub-optimal actions (Sutton and Barto, 1998), allowing better routing opportunities to be explored in a dynamic environment, however, at the expense of overhead.

The placement of this component is crucial as it dictates the design of the solution. Typically, this component can either be placed at newly-arrived devices that wish to take over the packet and become the next hop, or at the current hop that wishes to find a potential next hop.

In the former case, the packet will be voluntarily taken over and forwarded towards the destination. The destination may receive multiple copies of the same packet. This mechanism places most of the routing-related load at the more capable devices. It increases the data delivery probability as multiple copies of the same packet will travel towards the destination. This, however, also increases the network traffic.

In the latter case, the newly-arrived device advertises itself along with its ability to

deliver data to the destination. Taking advantage of the newly-arrived device is at the discretion of the current hop. This mechanism ensures that only a single copy of the packet reaches the destination, at the expense of slightly increasing processing load at the less capable devices. It reduces the overall traffic in the network, but also the data delivery probability.

The decision making component used in this thesis is split into two parts, such that the decision is partially taken at the newly arrived more capable mobile device and partially at the less-capable sensor node.

3.1.2 Performance Indicators

In order to take advantage of mobile devices while delivering sensor data from resource-constrained sensor networks, device differentiation and data delivery ability assessment mechanisms are required. A device's packet delivery ability can be divided into device's own *capabilities* dictated by the resources and roles that it possesses, and the device's *performance metrics* that cumulatively reflect the "goodness" of a device for delivering packets to their destinations.

Device capability reflects the tasks that the device can perform. A device that has an active long-range radio can send packets to Internet destinations. Similarly, a device with rechargeable or un-limited energy reserves can forward packets more frequently. Thus, the available devices can be categorized according to their capabilities such as command and control server, fixed gateway, mobile device and sensor node. A sensor node can be further classified as either cluster head or end device, where the end device is the leaf node of the cluster-tree network and does not allow other sensor nodes to consider it as their parent node while establishing paths towards the destination. A sensor node chooses a stationary parent node that can perform better than all other stationary nodes discovered during the radio channel scan.

Devices' performance metrics specify the suitability or capacity of a device to deliver packets to the destination. They will improve as a device successfully delivers packets

to their destinations. They not only depend on performance statistics, but also on the current environment, i.e., a device's neighbours. The performance metrics can be further divided into long-term and short-term components.

In terms of RL, this component can consist of a state-value function (e.g., a Q-value function), influenced by a number of factors (discussed below). Different weights can be associated with information from the current environment and past performance. In this thesis, a performance score is calculated for each neighbouring device based on its closeness to the destination, energy reserves, current load and past performance. The closeness of a device to the destination is depicted by a custom metric, i.e., *credentials*, that allows the routing protocol to infer direction in which the packet will travel to reach the destination. The performance score of each neighbour is used to determine a suitable data forwarder towards the destination.

3.1.2.1 Current state

The current state is influenced by the short-term component of the device's delivery ability. Factors such as remaining energy, distance to destination in terms of credentials, number of dependent devices and the number of packets transmitted contribute to the current state.

In a highly dynamic environment, acquiring the route health information is very costly. Therefore, a mobile device stores, updates and advertises its own delivery abilities. This strategy contributes in shifting the routing-related load towards more capable devices. Also, more collisions due to periodic beacon broadcast can be expected in a synchronized network, if un-synchronized mobile devices broadcast periodic beacons. For this reason, in this thesis, the mobile devices do not periodically broadcast beacons, rather they advertise their presence to the sensor nodes in their vicinity.

3.1.2.2 Rewards from feedback and past statistics

In terms of RL, rewards are received via feedback from the downstream nodes, as a device participates in data forwarding. The performance of devices used previously as next hops can be evaluated from these rewards, in terms of successful data forwarding. In this work, these rewards correspond to acknowledgement packets received from the downstream nodes upon successful delivery of a packet to the next hop. They influence both short-term and long-term data delivery abilities of a device. In case of delivery failure via a particular device, the performance metrics (particularly credentials) are re-evaluated based on a different device, influencing short-term component of performance metrics.

Past statistics can be recorded and used as the device participates in data forwarding, influencing the long-term component of the performance metrics. Successful deliveries to particular devices become part of these statistics. Information reliability, however, deteriorates with time.

In this thesis, it is assumed that the devices of the same type behave similarly. Because the resource-constrained sensor nodes have very limited memory, the past statistic about (un)successful forwarding to individual devices is not stored. Instead, the deliverability characteristics based on device type are stored, e.g., risk involved while forwarding via the parent device, child device or mobile device. These characteristics per device type are thus updated in real-time as devices participate in data forwarding towards the destination.

3.1.2.3 Neighbourhood

A device's neighbourhood's capabilities and credentials influence the short-term component of its delivery ability. Challenges associated with the discovery mechanism are discussed in the next section.

3.1.3 Neighbour discovery

This component is concerned with the neighbourhood discovery and the credential propagation.

A device can either influence its neighbourhood by periodically advertising its abilities, or be influenced by its neighbourhood by promiscuously listening to the packet transmissions in its vicinity. In the former case, a device will discover all neighbours, increasing the memory requirements of the protocol. In the latter case, a device will discover only those neighbours that are currently forwarding data and none if no data is currently transmitted.

As this work aims to reduce the energy consumption and communication load on resource-constrained sensor nodes, the former option of periodically advertising a device's delivery abilities by broadcasting beacons was chosen. This option allows the protocol to configure the inter-beacon interval without relying on continuous promiscuous listening, thus keeping the radio module in sleep mode for some portion of the time, and saving energy.

In order to discover newly-arrived more capable mobile devices to shift the communication load, either the resource-constrained sensor nodes or mobile devices need to scan the radio channel. Detailed discussion on the two available baseline options is carried out in Section 3.3.

In this work, the neighbourhood discovery is segmented into two phases, i.e., network formation and routing phases. The resource-constrained sensor nodes discover their neighbours during the network formation phase only. During this phase, the sensor nodes scan the radio channel in linearly increasing time intervals, until they find a neighbouring sensor node (or fixed gateway) that is capable of becoming their parent in order to connect them to the destination (command and control server). During the network formation phase, only stationary nodes are considered as potential parent devices. The network formation phase completes once the sensor nodes are connected.

During the routing phase, i.e., once the sensor nodes start generating sensor data, more capable mobile devices scan the radio channel and discover their neighbours, to shift some load from the resource-constrained sensor nodes. The newly-arrived devices synchronize with the waking period of the sensor nodes. After this synchronization completes, a newly-arrived mobile device advertises its presence to the sensor nodes, which in turn updates its performance metrics (particularly credentials) based on the mobile device, if the mobile device can become a better next hop than the sensor node's parent device (if any).

3.1.4 Risk Feedback

A device will acknowledge the reception of a data packet to the previous hop. Typically in RL, the feedback contains an estimated reward that is used to improve future action selections.

For simplicity, in this thesis, the reception of an acknowledgement alone is considered as a feedback at the previous hop. If the acknowledgement is not received after a certain time out, the risk of forwarding via the device is updated. This risk is maintained per device type instead of maintaining risk per individual device to improve the memory consumption at the resource-constrained sensor nodes.

3.2 Wireless Sensor Network Structure and Architecture

Device differentiation is incorporated in the design of the routing protocol, as the operations performed by the devices are divided according to the roles of the devices in the network. The network consists of four main types of devices as follows:

1. Centralized Command and Control Server.
2. Fixed Gateway.
3. Sensor Nodes.

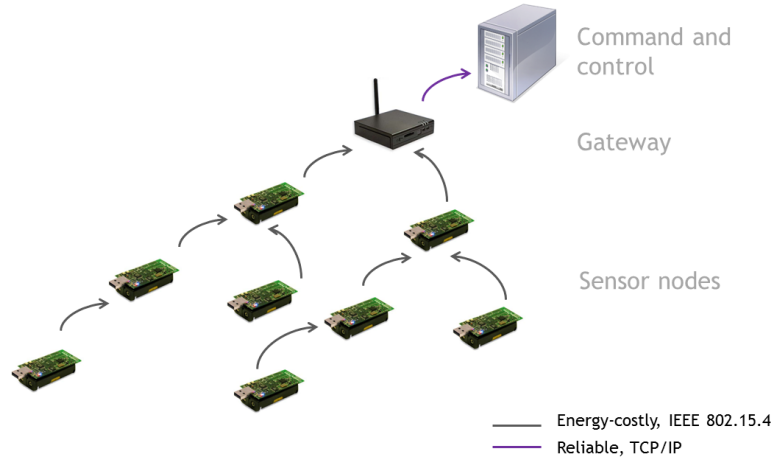


Fig. 3.2: Cluster Tree Network Topology.

4. Mobile Devices.

As discussed in Chapter 1, the command and control server is assumed to be the final destination of the sensor data, whereas the fixed gateway provides the bridge between the sensor network and the command and control server. The fixed gateway and command and control server are connected together using a reliable TCP/IP connection over the Internet or Intranet network. The fixed gateway also acts as the PAN (Personal Area Network) Coordinator for the WSN. All the sensor nodes initially connect to the fixed gateway via multi hop routes (Figure 3.2).

This network topology is formed dynamically to provide a fixed path, following which sensor data from all sensor nodes can reach the destination if the path exists. Even though the path provided by the aforementioned network topology is not completely reliable, it still provides a good mechanism to deliver packets to the destination. As the sensor nodes are battery operated, data forwarding via these paths is expensive because of the energy consumption due to data transmissions at the intermediary nodes routing data on behalf of other sensor nodes. As discussed in Chapter 1, nodes near the fixed gateway experience congestion due to the many packets arriving from multiple sources

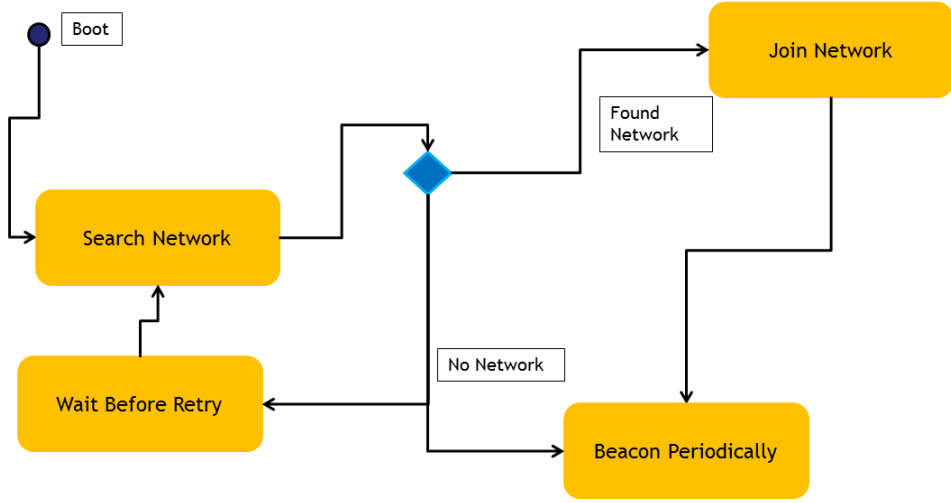


Fig. 3.3: Sensor Node's Operations at Boot Time.

that converge at the gateway. For this reason, extra-WSN mobile devices are used in this research to bypass intermediary sensor nodes. These extra-WSN mobile devices are also assumed to be connected to the command and control server via the Internet.

3.2.1 Wireless sensor network formation overview

At an abstract design level, when a sensor node is switched on, it searches for other sensor nodes in the network. The sensor node joins the network by connecting to a suitable neighbouring node, if other sensor nodes are found. After establishing a connection, the sensor node starts broadcasting periodic beacon messages. In the absence of a sensor network, the sensor nodes will broadcast beacons anyway, so that they can be discovered by mobile devices, which may provide a path to the destination. If sensor nodes cannot find another sensor node in order to join the network, they wait for some time before searching for neighbouring sensor nodes again. This abstract level of a sensor node's operation is shown in Figure 3.3.

The address of the command and control server is configured on the fixed gateway, so that the gateway can establish a connection with the server. As the fixed gateway is

assumed to be the PAN coordinator or the starting point of the network, it does not scan the radio channel in search of sensor nodes. Instead, the gateway starts broadcasting periodic beacons, as soon as it is switched on.

As the sensor nodes are resource-constrained devices in every aspect, it is absolutely imperative to keep the design of the mechanisms and their implementation simple. The problem of leveraging mobile devices for data forwarding becomes particularly challenging due to this reality. For this reason, the mechanisms employed during the network formation are also used to leverage mobile devices with minimal extensions possible on the sensor nodes.

3.2.2 Sensor node joining wireless sensor network

As shown in Figure 3.4, when a sensor node is switched on, it starts scanning the radio channel. If no neighbours are found, the sensor node starts a timer according to Equation (3.1), which depends on the previous value of the *nextScanAfter* variable, which is initially set to 0, and the *DefaultScanDuration* which is a constant value. At the same time, the sensor node also starts broadcasting periodic beacons. As discussed earlier in Section 3.2.1, the sensor node starts periodic beaconing so that other sensor nodes and mobile devices may be able to discover it. This allows the possibility to deliver data when a device capable of delivering data to the destination appears. When the timer expires, i.e., the timer event is triggered, the sensor node stops broadcasting beacons and scan the radio channel again. The periodic beacons are stopped during scanning, so that the scan process is not interrupted and beacons from all neighbours are received.

$$nextScanAfter_{(t+1)} = nextScanAfter_{(t)} + DefaultScanDuration \quad (3.1)$$

If neighbours are present in the vicinity of the sensor node, the sensor node selects a suitable neighbour that can potentially connect the sensor node to the destination.

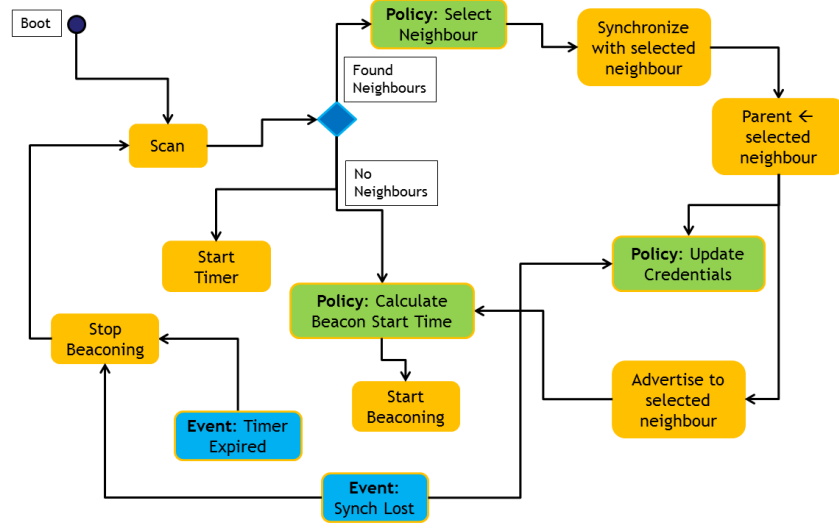


Fig. 3.4: Sensor Node Joining WSN.

Figure 3.4 also shows that if a suitable neighbour is selected then the sensor node synchronizes with the neighbour and then designates the selected neighbour as its parent node. This designation of a neighbouring node as parent, allows the sensor node to send data packets to the destination via the parent node, in essence providing a default path towards the command and control server via the fixed gateway. The sensor node then updates its own credentials based on the newly-selected parent node and sends an advertisement packet to the selected parent. After sending the advertisement, the sensor node starts broadcasting periodic beacon packets.

Before the sensor node starts broadcasting periodic beacons, the sensor node calculates the starting time of the beacons with respect to its parent (Appendix A.1).

If synchronization with the parent is lost, the sensor node updates its credential again. Next the sensor node stops broadcasting periodic beacons and scans the radio channel again.

While a sensor node is attempting to join the network, only stationary neighbours are considered. Mobile devices are not considered as potential parent, during network

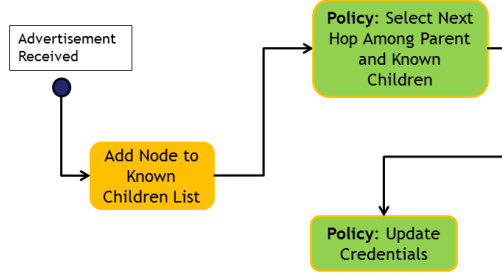


Fig. 3.5: Adding a node as a child.

formation, because they do not offer stable routes.

In Figure 3.4, blue boxes represent asynchronous events that are triggered independently of all mechanisms, the green boxes represent individual policies that are independent of the mechanism in which they are used, and the orange boxes represent mechanisms. This technique of separating policies and mechanisms is inspired partly from Levin et al. (1975). Each of the policies will be discussed separately.

3.2.3 Adding a node as a child

Each sensor node maintains a list of all known child devices. When a sensor node receives an advertisement from another device in the network, it simply adds the advertising device into this list. The children of a sensor node are classified as dependent sensor nodes and mobile devices, such that the mobile devices do not depend on the sensor node to establish path towards the destination.

By default, the parent of a sensor node is configured as the next hop of any data packet that the sensor node wishes to send to the destination. However, Figure 3.5 shows that upon receiving an advertisement, the sensor node re-evaluates its choice of the next hop. In doing so, the sensor node compares all the neighbouring devices in its list with the designated parent node, and selects the most suitable device according to the selection policy.

This mechanism allows the sensor node to use one of its children, if it can perform

better than its parent. Such scenario can occur if a mobile device arrives in range and offers its services to the sensor network for data forwarding.

If the sensor node selects a device from its list, other than its designated parent, the sensor node updates its credentials based on the selected device. However, the selected device is not designated as the sensor node's parent this time, because the sensor node itself is the parent of the selected device. Designating the selected device as parent, in this case, will result in cyclic dependency. In addition, the non-parent selected device, represents temporarily better path towards the destination.

3.3 Discovery Mechanism

Initially the sensor nodes (SNs) scan the designated radio channel(s) to discover other stationary SNs and set up a fixed network, if possible. This enables the sensor nodes to form fixed routes towards the destination (command and control server) via the fixed gateway. A hierarchical network topology can be formed using this mechanism, similarly to Tennina et al. (2011a).

In order to take advantage of mobile devices to outsource data forwarding and processing load to these devices, the mobile devices and the sensor nodes must discover each other quickly. For discovery to work, one device must transmit regularly and the other device must listen (discover).

In wireless sensor networks (WSNs), the deployed SNs have limited energy reserves (Chapter 1). Therefore, the discovery mechanism must be energy efficient, especially on sensor nodes. Additionally, the SN part of the discovery mechanism should be light weight, because SNs have very limited ROM/RAM space. In contrast, mobile data collectors (MDCs) have significantly better energy reserves, memory, and processing and communicating capabilities. In addition to being energy efficient and light weight, the discovery mechanism has to be fast enough to allow the sensor nodes to effectively communicate with mobile data collectors, despite their potentially high mobility speed.

This section discusses the network formation mechanism available in the IEEE-802.15.4 standard (Section 3.3.1) that becomes the basis of the proposed discovery mechanism. The majority of the existing approaches place the discovery mechanism at the resource-constrained sensor nodes, such that the stationary sensors listen and the MDCs beacon periodically. Since listening consumes more energy than transmitting in typical WSNs (Chipcon, 2006), it is hypothesized that the listening duration should be reduced at the sensor nodes. Inspired by Liang et al. (2008), in the approach presented in this thesis, the sensor nodes periodically beacon and the mobile devices periodically listen for beacons, as a first step, but exploits synchronized nature of the network. Sections 3.3.2 (conventional discovery mechanism) and 3.3.3 (alternative discovery mechanism) initially test the hypothesis by analysing the listening duty cycle at the sensor nodes and the discovery time, concluding that the alternative approach presents a more deterministic listening duty cycle at the sensor nodes while having same discovery time. Section 3.3.4 presents the proposed “Inverse Location Aware Discovery” mechanism that builds on and refines the existing alternative approach by incorporating location awareness, such that the radio scan at the MDC is triggered when sensor nodes are expected in its vicinity. In addition to being energy efficient, this variant is also expected to exhibit faster discovery times.

The discovery mechanisms, discussed in this section, can be implemented using any MAC protocol that incorporates periodic beacons, scans and inactive intervals.

3.3.1 Device discovery in IEEE-802.15.4 (2006) (Beacon-enabled mode)

In the standard IEEE-802.15.4 beacon-enabled mode, a device can discover other devices by scanning the designated radio channel(s). This mechanism assumes that the beacon order (BO) parameter has the same value on both devices. The beacon order (BO) dictates the interval between two consecutive beacons, i.e., the (inter-)beacon interval (BI). The device to be discovered, transmits beacons on a designated radio channel every BI, and listens for data from children devices during the active portion of the superframe.

The active portion of the superframe is termed as the superframe duration (T_{sfd}) and is controlled by varying the superframe order (SO) parameter. A discovering device scans the radio channel(s) for a time period (scan duration) (IEEE802.15.4, 2006). The BI, the superframe duration (T_{sfd}) and the scan duration (T_{scn}) are given by Equations (3.2), (3.3) and (3.4) respectively, for $0 \leq SO \leq BO \leq 14$, where SO is superframe order and aBaseSuperframeDuration (BSD) equals 960 symbols (IEEE802.15.4, 2006). The actual value of aBaseSuperframeDuration (BSD) in terms of time, however, depends on the clock precision of the hardware. Lu et al. (2004) derive the value of BSD to be 0.01536 seconds in their simulations.

$$BI = BSD * 2^{BO} \quad (3.2)$$

$$T_{sfd} = BSD * 2^{SO} \quad (3.3)$$

$$T_{scn} = BSD * (2^{BO} + 1) \quad (3.4)$$

All beacons received during the scan duration are reported to the network layer, which then selects a device based on the received beacons and synchronizes with the selected device.

During the network formation, a device is typically set up to scan the radio channel(s) periodically, until it detects a suitable device. After synchronization, the selected device is designated as the parent. The scanning process is triggered again if the synchronization with the parent device is lost.

3.3.2 Stationary SNs periodically re-scan

In existing approaches where Mobile Data Collectors (MDCs) are present, the scanning process cannot be stopped after synchronization because doing so would hinder the discovery of new MDCs. Generally, this problem is mitigated by scanning the radio

channel(s) periodically every *BackOff* interval, allowing the devices to adapt to the changing environment. In this approach, the periodic scanning functionality is placed at the un-attended SNs. Each MDC periodically broadcasts beacons, which are detected in the periodic re-scans of the SNs. Once a suitable MDC is discovered, the SN can synchronize and establish communication with the MDC.

For this baseline discovery mechanism, the listening duty cycle¹ at the SN can be given by Equation (3.5) when no suitable device is detected. However, if a suitable device is selected, i.e., the scan was successful, then the listening duty cycle can be given by Equation (3.6), where T_{sync} is the time taken to synchronize. Therefore, on average, the listening duty cycle can be estimated by Equation (3.7), where γ represents the proportion of time where a SN chooses to synchronize with a MDC.

$$D_{(NotFound)} = \frac{T_{scn}}{T_{scn} + BackOff} \quad (3.5)$$

$$D_{(Found)} = \frac{T_{scn} + T_{sync}}{T_{scn} + T_{sync} + BackOff} \quad (3.6)$$

$$D_{(listen)} = \gamma * D_{(Found)} + (1 - \gamma) * D_{(NotFound)} \quad (3.7)$$

The time taken by the sensor node to discover a MDC (T_D) can generally be given by Equation (3.8), where δ represents any software delays. The lower the value of T_D , the faster a MDC is discovered. The *BackOff* parameter controls the maximum discovery time, whereas from Equation (3.4), the BO parameter affects the minimum discovery time, because T_{scn} theoretically remains constant. T_{sync} is variable and depends on when a beacon is received after the scan was completed. The maximum value of T_{sync} approximately equals T_{scn} .

$$T_D = BackOff + T_{scn} + T_{sync} + \delta \quad (3.8)$$

¹Proportion of time a device listens.

3.3.2.1 Detailed Design

The MDC starts broadcasting periodic beacons as soon as it is switched on. The periodic re-scanning is scheduled every *BackOff* period at the sensor node. The *backoff* period starts at the end of the previous scan cycle. The scan cycle consists of a scanning interval (T_{scn}), followed by the selection of the appropriate MDC, followed by synchronization time (T_{sync}). If the scan is successful, i.e., an appropriate MDC is selected, an advertisement (ADV) message is sent to the MDC by the SN in T_{adv} time, thus Equation (3.8) can be translated into Equation (3.9). The scan cycle ends after ADV is sent (if MDC is selected) or at the end of T_{scn} . The scan cycle begins again when this *backoff* timer expires.

If the stationary nodes and MDCs are discovered in the same scan cycle, then priority will be given to the stationary node if no parent device is previously selected. Otherwise, a MDC will be selected in the scan cycle if it can perform better than the parent device². This enables network formation without depending on the availability of MDCs.

$$T_D = BackOff + T_{scn} + T_{sync} + T_{adv} + \delta \quad (3.9)$$

The minimum time taken to discover a MDC ($T_{D_{min}}$) is achieved when the MDC arrives in range just as the scan cycle starts and a beacon from the MDC is received at the beginning of the synchronization period (Figure 3.6). Therefore, it can be formulated as Equation (3.10), such that both BackOff and T_{sync} are almost zero.

$$\lim_{T_{sync} \rightarrow 0} T_{D_{min}} = T_{scn} + T_{adv} + \delta, \text{ BackOff} \approx 0 \quad (3.10)$$

The maximum time to discover a MDC is expected when the MDC arrives in range right after the scanning cycle at the SN completes, and when the next scan cycle begins, beacons are received at the end of the synchronization periods, resulting in full backoff

²Parent device is a stationary node that can perform better than all other stationary nodes discovered in the scan cycle.

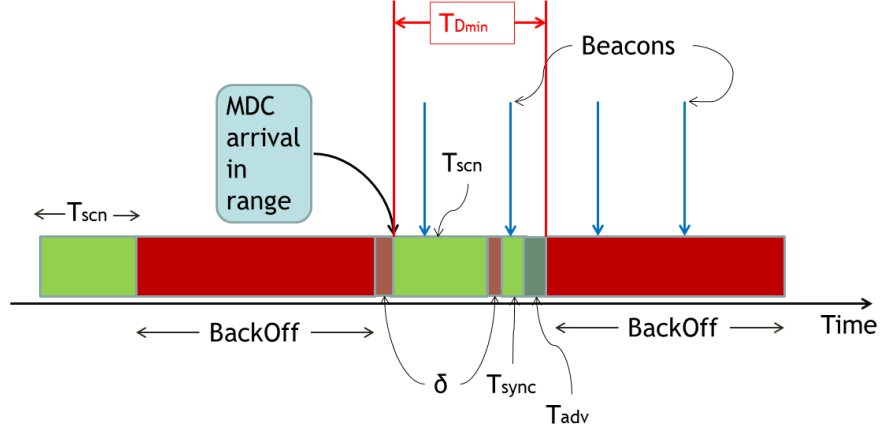


Fig. 3.6: Representation of achieving $T_{D_{min}}$

and synchronization period (Figure 3.7). Thus, $T_{D_{max}}$ can be formulated as Equation (3.11), where $T_{sync} \approx T_{scn}$.

$$T_{D_{max}} = BackOff + 2 * T_{scn} + T_{adv} + \delta \quad (3.11)$$

Hence, the time to discover a MDC can be theoretically given by the interval: $[T_{scn} + T_{adv} + \delta, BackOff + 2 * T_{scn} + T_{adv} + \delta]$.

3.3.2.2 Discussion

From the author's experience, a few note-worthy observations, related to this conventional discovery mechanism, are outlined below.

If more than one beaconing mobile devices are present in the same collision domain, then their periodic beacons may collide, resulting in the failure of their discovery by the periodically scanning sensor nodes.

In addition, if this discovery mechanism is implemented in an established network, then every time a sensor node starts scanning, it will switch off its periodic beaconing in order to avoid interruption during scan. On resource-constrained sensor nodes, while

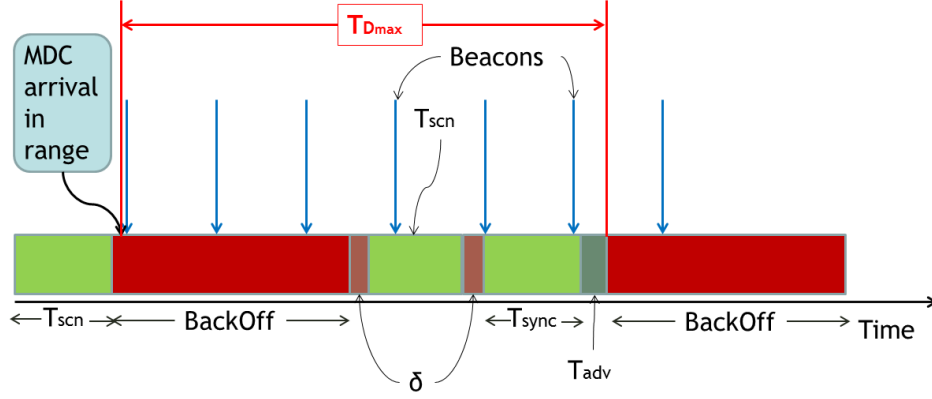


Fig. 3.7: Representation of obtaining $T_{D_{max}}$

scanning the radio channel, the periodic beaconing has to be temporarily switched off so that the radio channel scan does not get pre-empted, because these resource-constrained sensor nodes are single threaded, single processor systems. Due to this, any sensor nodes connected to this sensor node will lose their connectivity, triggering scan at the child sensor nodes as well, resulting in overall higher energy consumption.

3.3.3 Mobile devices periodically re-scan

In this mechanism, the unattended sensor nodes periodically broadcast beacon frames, every BI. The MDCs scan the designated radio channel(s) periodically after every *BackOff* period, such that $BackOff \geq BI$, allowing time for data transfer. In this way, the discovery procedure is shifted from less capable SNs to more capable MDCs, fulfilling one part of the overall objective of this thesis of shifting most of the computation and communication load to more capable (mobile) devices.

Because listening consumes as much or more energy than transmissions, especially for CC2420 (Chipcon, 2006) (the radio module used in many SNs including TelosB (Crossbow, 2005)), this mechanism is expected to reduce the energy consumption at SNs by reducing the listening duration over time, for appropriate values of SO and BO

parameters, while making it possible to achieve faster discovery time by reducing BackOff at the MDC. In fact, CC2420 consumes 19.7 mA in receive mode, while in transmission mode the current consumption ranges from 8.5 mA to 17.4 mA for the transmission power ranges of -25 dBm to 0 dBm.

Because no extra optimizations are performed, e.g., adaptive discovery frequency, the discovery time is expected to be similar to that of the baseline protocol (Section 3.3.2) for same BackOff and BO parameters. Thus Equation (3.8) also applies in this case.

For this alternative discovery mechanism, the listening duty cycle at the SN can be given by Equation (3.12), as SNs only listen during the superframe duration.

$$D_{(listen)} = \frac{BSD * 2^{SO}}{BSD * 2^{BO}} = \frac{2^{SO}}{2^{BO}} \quad (3.12)$$

3.3.3.1 Detailed Design

In this mechanism, after the fixed network formation, the SNs do not scan any further and start broadcasting periodic beacons every BI.

In order to discover SNs, the MDCs could have listened promiscuously to the data transmissions by the SNs. However, if there are no existing routes, the SNs may not transmit anything. Therefore, the MDCs will be unable to discover SNs and unable to offer routes to them or consume their data. For this reason, periodic beaconing mechanism provided by IEEE-802.15.4 was used instead, such that the MDCs scan periodically every *BackOff* period, and the SNs broadcast periodic beacons. If a MDC discovers a SN, it synchronizes with the selected SN, and sends an ADV packet to the SN. The SN can start using this MDC upon receiving the ADV packet.

In this mechanism as well, the time taken to discover stationary SNs can be given by Equation (3.9). The variations of Equation (3.9) and the discovery interval $[T_{scn} + T_{adv} + \delta, BackOff + 2 * T_{scn} + T_{adv} + \delta]$ are also valid.

3.3.4 Inverse Location-Aware Discovery

This thesis proposes the “Inverse Location-Aware Discovery”. This discovery mechanism is developed as a variant of the mechanism discussed in the previous section. This mechanism relies on the assumption that the location of the stationary WSN is known by the command and control server and shared with the MDCs. Thus, the MDC will only start a radio scan when it expects sensor nodes in its vicinity.

As the sensor nodes beacon periodically, their listening duty cycle can be given by Equation (3.12). The listening duty cycle at the sensor nodes can be controlled by varying SO and BO parameters.

This discovery mechanism is expected to yield much lower discovery intervals than the mechanisms discussed in Sections 3.3.2 and 3.3.3, while achieving energy conservation at the sensor nodes.

For this mechanism and the one discussed in the previous section, $\frac{2^{BO}}{2^{SO}}$ beaconing devices can be deployed in a single collision domain, using the time division beacon scheduling technique (Tennina et al., 2011b). The stationary network can be further scaled by avoiding beacon collisions using a spatially diverse deployment. The number of mobile devices can also be easily scaled up due to absence of heavy traffic originating from the mobile devices.

3.4 The STERIOD Routing Protocol

In typical wireless sensor networks (WSNs), the sensor data is collected at a centralized destination, from where it can be propagated to individual users, e.g., in the EMMON project (Tennina et al., 2011a). As discussed in Chapter 1, this thesis assumes that the destination for the sensor data is a centralized command and control server, which is connected to the WSN via an Internet or Intranet connection. As discussed in Chapter 1, Section 1.7.1, the WSN itself, is assumed to be organized in a cluster-tree network topology. The design of the routing protocol aims to reduce congestion at critical nodes

of the network, e.g., nodes that are deployed closer to the fixed gateway. In addition, the routing protocol aims to reduce the energy consumption at the nodes that forward sensor data on behalf of other nodes and also provide fault tolerance in case of network partitions (Chapter 1, Section 1.5). As discussed in Chapter 2, none of the existing work fulfils the requirements and achieves the goals set in this thesis.

This section discusses performance indicators (Section 3.4.1), credential and risk feedback (Section 3.4.2), and decision making components (Sections 3.4.3, 3.4.4 and 3.4.5).

3.4.1 Performance Indicators

In order to improve performance in terms of energy consumption, congestion and latency, independent performance indicators are used. These indicators include remaining energy level, number of dependent children, number of packets transmitted in the last minute and device credentials. These performance indicators are included in the beacon packets of each of the sensor nodes, in addition to the generic information about the device (e.g., frame control, addressing fields, superframe specification, guaranteed time slot information) (IEEE802.15.4, 2006). These performance indicators are used by the decision making component in separate capacities.

3.4.1.1 Energy

This performance indicator reflects the remaining energy of a device at any given instant.

As the fixed gateway is assumed to be line-powered, its remaining energy is fixed at a constant value.

The remaining energy of a sensor node is obtained with the on-board voltage sensor, as the voltage is a direct indicator of the remaining energy. The voltage sensor reading of the sensor node ranges between zero and 4096, such that $Voltage = \{0, 1, 2, \dots, 4096\}$ and $Voltage \in \mathbb{N}$.

The remaining energy of a mobile device is obtained in terms of percentage, ranging

between zero and hundred. For mobile devices, the remaining energy percentage is a real number. However, for calculation at the sensor nodes, this value will be converted to a whole number (non-fractional number).

Due to the disparity between the range of remaining energy at different types of devices, the value of remaining energy is normalized. The normalized energy of a device is given by Equation (3.13).

$$NormalizedEnergy(d) = \frac{Energy(d)}{MaximumEnergy(Type(d))} \times NormalizedUpperBound \quad (3.13)$$

In Equation (3.13), Type(d) refers to the type of the device and NormalizedUpperBound is a constant that brings the remaining energy ratio to a normalized value.

The energy value always decreases as the device is used, unless the device is recharged.

3.4.1.2 Credentials

Credentials are the main performance indicator for any given device. The closeness of any node to the destination (centralized command and control server) is given by the credential value, which reflects how good a device is to reach the destination in the present network topology. The credentials also give a sense of direction in which the packet will travel to reach the destination, from any given node. Therefore, as long as the network topology remains same, the credential value for each device will also remain the same. The credential value is assumed to vary between zero and hundred, such that $Credential = \{0, 1, 2, \dots, 100\}$ and $Credential \in \mathbb{N}$.

The credential value of a device is initialized to zero, if it has no connection to the destination. Upon connecting to a node closer to the destination, the device updates its credentials according to the potential next hop's credentials. The device will update its credential according to Equation (3.14), where Cr(d) represents the calculated credentials of device d , Cr(d-1) represents credentials of the potential next hop device, and ξ

represents a degradation factor per hop.

$$Cr(d) = Cr(d - 1) \times (1 - \xi) \quad (3.14)$$

The gateway has the highest possible credential value, because it is assumed to be directly connected to the centralized command and control server. If a mobile device is also connected directly to the command and control server using a long range radio, then it will also have the highest possible credential value.

3.4.1.3 Number of dependent children

The number of dependent children reflects the number of sensor nodes that have advertised to a particular device. This performance indicator gives an estimate of congestion on a given device, meaning how many (one-hop) devices are expected to send their sensor readings to this device to be forwarded to the destination. The number of dependent children only includes stationary nodes that have advertised to a particular device, as mobile devices are not assumed to be the source of sensor data in this research.

3.4.1.4 Number of packets transmitted in the last minute

This performance indicator estimates the amount of outbound traffic on any sensor node. As only the latest conditions are relevant for decision making, the sensor nodes only include the number of packets transmitted in the last minute, in their beacons. This metric allows a run-time estimate of the traffic congestion at a particular node.

3.4.1.5 Device location

The device location is not an actual performance indicator, rather it allows the mobile device to identify the devices that are in proximity to each other. The two dimensional coordinates of the sensor node are included in its periodic beacons, so that when the sensor node selection mechanism is triggered at the mobile device, the mobile device can

approximate which sensor nodes are in range of each other and which sensor nodes may be transmitting to which other sensor nodes without relying on promiscuous listening. The transmission range of the sensor nodes is assumed to be known a priori as only bi-directional (symmetric) links are considered, and the transmission range is assumed to be constant throughout the sensor network irrespective of the device type (Chapter 1, Section 1.7.1).

3.4.2 Credential and Risk Feedback

When a mobile device arrives in the vicinity of sensor nodes, if it successfully discovers and selects a sensor node, it sends an advertisement to the selected sensor node. This advertisement packet allows the sensor nodes to take advantage of the mobile device while forwarding packets to the destination.

Upon receiving an advertisement message, a sensor node checks if the advertising node is better than its own selected parent. If the advertising node is in fact better, then the sensor node selects the advertising node as its potential next hop and update its credentials again with respect to the advertising node according to Equation (3.14).

If the percentage difference of the updated credential value with respect to the previously held credential value is greater than the degradation factor (i.e., $PercentDifference(Cr(d), Cr'(d)) > \xi$), then the sensor node also sends a new advertisement to its selected parent. The percentage difference of credentials is calculated by Equation (3.15) and represents the percentage by which the credentials have changed.

$$PercentageDifference(Cr(d), Cr'(d)) = \frac{|Cr(d) - Cr'(d)|}{Average(Cr(d), Cr'(d))} \times 100 \quad (3.15)$$

In Equation (3.15), $Cr(d)$ represents the current (updated) credentials of the device, and $Cr'(d)$ represents the previous credentials of the device. This refers to the relative change in the credentials, such that the previous credential value is taken as reference. As the credential update mechanism is only concerned with deciding whether a new ad-

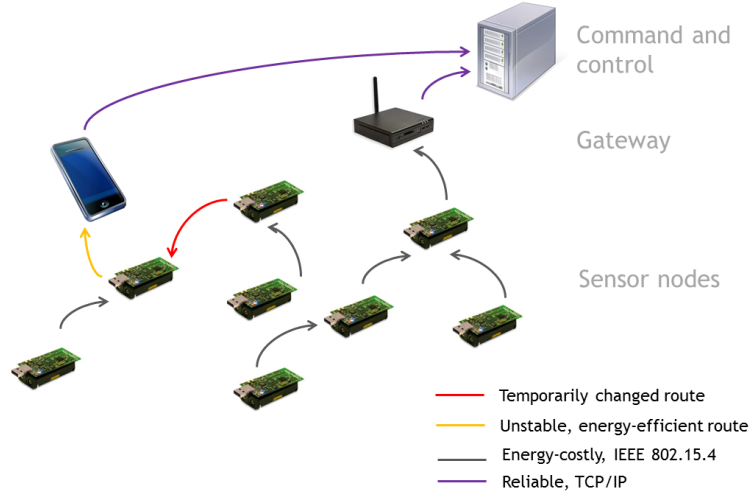


Fig. 3.8: Cluster tree network topology modified due to presence of mobile device.

vertisement should be sent to the parent node, the absolute value of percentage difference is used rather than actual percentage difference.

The new advertisement, sent to the parent device by a sensor node, contains the updated credential value of the sensor node. The mechanism of re-advertising to the selected parent node allows the parent node as well to take advantage of the mobile device indirectly. Figure 3.8 shows the change in the path taken by data packets when compared to Figure 3.2, such that packets generated in one branch of the tree are shown to be forwarded by the mobile device and not via the fixed gateway. The network connections, however, remain the same. That is, the network structure will still be the same.

In addition, the sensor nodes also update their credentials periodically with respect to their parent nodes (from the parent's beacons), so that any change in the credentials of the parent also affects the credentials of the child sensor nodes.

If the mobile device moves away, the sensor node notices the connection break when the next packet is forwarded. Upon connection break, the sensor node updates its credentials again according to Equation (3.14) with respect to the best neighbour (possibly

parent). The sensor node then checks if the percentage difference between its new credentials and its previous credentials has changed beyond the degradation factor. If this condition is met, then the sensor node again re-advertises to its parent, and the routes will return to their original state (Figure 3.2).

In the case when the mobile device moves away, the sensor node only notices a connection break when its transmission to the mobile device fails. For this reason, the sensor node keeps an additional performance indicator to keep track of the devices due to which it lost packets. This indicator is referred to as ‘*Risk*’. As the memory on resource-constrained sensor nodes is limited, it is difficult to keep track of all devices due to which the sensor node has lost packets. For this reason, the sensor node instead records ‘*Risk*’ per device type. Risk simply refers to the ratio of packets lost versus total packets sent via a certain type of device, given by Equation (3.16).

$$Risk(Type(d)) = \frac{PacketsLost(Type(d))}{PacketsSent(Type(d))} \quad (3.16)$$

The risk can only be calculated after trying different devices, and is not known a priori. In addition, the risk is associated with the next hop that a sensor node takes. Risk is not explicitly broadcasted in the beacon frames, rather it impacts the credentials, because just like credentials, risk too will only change if and when the network structure changes. For these reasons, Equation (3.14) can be re-written as Equation (3.17).

$$Cr(d) = Cr(d-1) \times (1 - \xi) - Risk(Type(d-1)) \quad (3.17)$$

In the cluster-tree network topology considered in this thesis, maintaining risk for types of devices becomes a natural and simplified choice because every sensor node has only three types of neighbours:

1. Parent node of a sensor node.
2. Dependent children of a sensor node.

3. Mobile devices connected to a sensor node.

In Equation (3.17), the risk associated with the type of selected next hop is subtracted from the credentials of the selected next hop (after degradation) because risk will have a negative impact on the performance of the sensor node.

3.4.3 Decision Making

The decision making component is divided into three parts. Each of these parts can be controlled independently.

1. **Parent Selection:** It is executed when a sensor node is joining the network and is used to decide which of the neighbouring devices is expected to provide a better route to the destination.
2. **Route Selection:** It is executed when a data packet is to be forwarded to the destination and is responsible for selecting the best candidate for the next hop, i.e., the node that is expected to forward data to the destination most efficiently.
3. **Sensor Node Selection by Mobile Device:** It is responsible for the selection of a sensor node by the mobile device. The mobile device chooses a sensor node that is most suitable to consume services offered by a mobile device.

This component is divided into two sub-components, such that one sub-component is implemented at the sensor nodes (Section 3.4.4) that consists of parent selection and route selection parts of the main decision making component, and the other sub-component is implemented at the mobile device (Section 3.4.5) that executes the sensor node selection part of the main decision making component.

3.4.4 Decision Making at Sensor Nodes

The parent and route selection parts of the decision making component are executed on the resource-constrained sensor node, therefore they are kept very simple. This strategy

is expected to reduce the complexity of the sensor nodes, and will shield them from performing complex calculations. In order to reduce complexity at the sensor nodes, both of these parts of the decision making component are greedy, i.e., the node with the highest score is selected.

3.4.4.1 Parent selection

The parent selection is only executed during the network formation phase, i.e., when a sensor node is started and it is joining the network. This only takes stationary devices that have a non-zero credential value into consideration. This design decision is taken so that a stable network can be established without depending on the mobile devices. The stable network allows the sensor nodes to send sensor readings to the destination even in the absence of mobile devices.

The parent selection policy calculates a score for each of its neighbouring (stationary) devices and greedily selects a parent as the node that has the highest score.

For parent selection, the score is based on the normalized energy, the credentials of the next hop and the number of dependent children of the next hop. In order to reach the destination quickly, the credentials of the next hop are used as one of the decision metric parameters. The normalized energy is used to make energy-aware decisions. The number of dependent children of the next hop are considered in the decision metric to select a parent device that is expected to have comparatively less traffic than other neighbours.

The score is calculated based on the performance indicators broadcasted in the beacon packets by the neighbouring devices, as shown in Equation (3.18).

$$Score(d) = \omega \times NormalizedEnergy(d) + (1 - \omega) \times Cr(d) - \eta \times Dependents(d) \quad (3.18)$$

In Equation (3.18), ω controls the weight between a neighbour's credentials and its normalized energy, resulting in an average value of the two parameters. The normalized

energy is calculated from Equation (3.13) taking the advertised energy present in the beacon as input. As the risk associated with the neighbours and the number of packets transmitted in the last minute, are not known during network formation, they are not used in calculating the score.

In Equation (3.18), $Dependents(d)$ gives the number of dependent children of the neighbouring node. This metric is used, so that congested nodes with more children can be avoided. This means that if two neighbours have same amount of normalized energy and credentials, then the neighbour with less dependent children will be favoured. The weight η associated with dependent children gives control over the congestion in the network, because neighbours with more children will be more congested (this is especially true in hierarchical-tree based network topology).

Upon selection of a parent node, the sensor node updates its credentials based on the parent using Equation (3.17), however, in the absence of risk, the credentials are updated using Equation(3.14).

3.4.4.2 Route selection

In most cases, this part of the decision making component is executed after the network formation is complete, unless a network partition occurs and sensor data is present to be sent to the destination.

In the absence of mobile devices, the only possible route to the destination is created via a device's parent node, selected by the parent selection part (Section 3.4.4.1). Only when a mobile device is present in the vicinity of the sensor network, its effect will be propagated in the network tree. As the mobile device advertises itself to the sensor node selected by the mobile device, from the network structure's perspective, the mobile device becomes a child node of the sensor node.

As discussed in Section 3.2.3, upon reception of an advertisement packet from the mobile device, a sensor node executes this policy to select the next hop among its parent and all of its children (including the mobile device).

While calculating a score for each device, the normalized energy reserves and the credentials are considered to reach the destination quickly and energy efficiently. In addition, the risk associated with the device type is subtracted from the score to identify the next hop with the least risk involved while delivering sensor data to the destination, in order to increase chances of successful data delivery. In order to reduce congestion, a real-time estimate of the number of packets transmitted in the last minute is also used, if the score is being calculated for a sensor node. This number of packets transmitted in the last minute is not required to calculate the score for a more capable device (mobile device or fixed gateway), as these devices are considered to be able to handle large number of data packets.

This policy calculates a routing score according to Equation (3.19), using the independent performance indicators discussed in Section 3.4.1. These indicators are also associated with each child node as they are present in the advertisement packet as well. The risk, however, is maintained by the sensor node itself and is not present in either the advertisement or beacon packets.

$$Score(d) = \begin{cases} \omega \times NormalizedEnergy(d) + (1 - \omega) \times Cr(d) \\ -Risk(Type(d)) & \text{if } d \text{ is mobile/gateway} \\ \omega \times NormalizedEnergy(d) + (1 - \omega) \times Cr(d) \\ -Risk(Type(d)) - \eta \times PacketsTxInLastMinute & \text{if } d \text{ is sensor node} \end{cases} \quad (3.19)$$

If Equation (3.19) is compared to Equation (3.18), there are three main differences.

1. The risk associated with a particular device type is used. This risk was not available previously during network formation while selecting a parent node.
2. If the score for a mobile device or fixed gateway is calculated, the number of dependent children or the number of packets transmitted in last one minute are

not applicable. This is due to the fact that the mobile device or fixed gateway have higher resources and thus all received packets are assumed to be accepted and forwarded to the destination by these devices.

3. If the score for a sensor node is calculated, then instead of the number of dependent children metric, the number of packets transmitted in the last minute is used. This is done so that the most recently known run-time estimate of transmission load is used for the resource-constrained sensor nodes, in order to reduce incoming traffic on such nodes.

Any packets arriving at the mobile device or the fixed gateway are simply forwarded to the command and control server using TCP/IP communication model by using configured IP address of the command and control server.

Because this policy is executed on the resource-constrained sensor nodes, a simple greedy selection is used. This means that the node with the highest score is selected. However, due to using a greedy selection policy and the presence of the risk performance metric, it is possible that at some stage the risk associated with mobile devices may grow large due to their mobility. If this situation occurs, then the mobile devices will not be selected any more. In order to allow continued usage of mobile devices, the sensor node could select a mobile device (if present) at random occasions, even if its score is much less than the rest of the nodes. This mechanism will be somewhat similar to ϵ -greedy action selection policy in reinforcement learning (Sutton and Barto, 1998). Such a mechanism, however, was not implemented in this research.

3.4.5 Decision Making at Mobile Device

As the mobile devices are the source of uncertainty and dynamism in the environment, it is natural to keep the majority of the decision making on the mobile devices themselves, as they are most aware of their behaviour and current state at any given time, and therefore, can take better informed decisions. In addition, it was hypothesized in Section 3.3

that if a mobile device discovers the sensor nodes by scanning the radio channel, the sensor nodes become more energy-efficient while keeping the discovery mechanism more reactive towards the sensor network. For these reasons, in this thesis, the mobile device is responsible for discovering sensor nodes in its vicinity by scanning the radio channel using Inverse Location-Aware Discovery (ILAD) mechanism, and then selecting one of the sensor nodes found in the scan.

In this case, the mobile device can only communicate with a single sensor node directly at a time. Therefore, the mobile device needs to select the sensor node carefully, in line with the objectives to be achieved.

This section first discusses the four objectives to be accomplished while offloading communication load away from resource-constrained sensor nodes (Section 3.4.5.1) and their prioritization (Section 3.4.5.2). After this, the actions to achieve each of the objectives are identified in Section 3.4.5.3 and Section 3.4.5.4 shows how to take a particular action. Section 3.4.5.5 discusses how network conditions dictate the achievement of an objective. Section 3.4.5.6 shows how all these parts are compiled together on the mobile devices in terms of strategies to achieve each objective.

3.4.5.1 Objectives

The objectives of the sensor node selection policy by mobile device are as follows:

1. **Fault Tolerance:** If sensor nodes exist in the vicinity of the mobile device, such that they are not connected to the network and cannot reach the final destination (the Command and Control Server), then the mobile device may become a bridge between such sensor nodes and the final destination, in order to increase data delivery ratio.
2. **Energy Conservation:** If sensor nodes exist in the vicinity of the mobile device, such that the sensor nodes are low on energy budget, then the mobile device may try to divert traffic away from such sensor nodes in order to reduce the traffic being

forwarded by such sensor nodes and in turn conserve energy on those sensor nodes.

3. **Congestion Reduction:** If sensor nodes exist in the vicinity of the mobile device, such that a high volume of traffic is arriving at these sensor nodes, then the mobile device may try to divert traffic away from such sensor nodes, in order to reduce congestion at them.
4. **Latency Reduction:** If sensor nodes exist in the vicinity of the mobile device, such that the sensor nodes are too far from the destination and the packets originating or being forwarded by these sensor nodes take a long time to reach the final destination, then the mobile device may provide a shorter/faster route by reducing the number of hops to the final destination.

In this thesis, critical sensor nodes are considered to be those on which many other nodes rely to forward their packets towards the destination. In a hierarchical sensor network topology (as considered in this thesis), the critical nodes are those that are deployed closer to the fixed gateway. However, in the presence of mobile devices, the sensor nodes that are closer to these mobile devices can also become critical nodes, because a lot of packets can be routed via these nodes.

The conventional view of congestion (Wan et al., 2003) mainly focuses on congestion on a receiving node due to channel contention. However, as this thesis assumes a hierarchical cluster-tree topology and uses IEEE802.15.4 (2006), contention-based congestion will be minimal. In addition, Guaranteed Time Slots (GTS) (IEEE802.15.4, 2006) can be utilized to provide contention-free communication when data is sent to the parent node from multiple child nodes. During the Contention Access Period (CAP) of IEEE802.15.4 (2006), the child nodes can still communicate with the parent node by dividing their transmission slots according to their unique address, hence reducing the chances of collision, as described for broadcasting beacons in Appendix A.1.

In this thesis, congestion refers to the amount of traffic or packets received/transmitted at an intermediary node (cluster head or router) that it will have to forward on behalf

of other sensor nodes. The packets forwarded by such a congested node will amount to increased energy consumption on that node, in addition to increasing latency for the packets due to re-transmissions and processing of input buffer queue respectively. In such a case, packets may be dropped or not forwarded by such congested nodes due to a buffer overflow or reduced energy level, in turn decreasing delivery ratio. Thus, packets transmitted in the last minute, provide a run-time measure to detect such congested nodes. This metric, included in the periodic beacons, allows the mobile device to identify sensor nodes experiencing a high volume of outgoing traffic at run-time and only reflects the most recent state at any given sensor node.

3.4.5.2 Objective prioritization

Because there are multiple objectives that need to be accomplished, and each of these objectives may be accomplished by selecting a different sensor node, the objectives mentioned above are prioritized according to their implications on the network. The fulfillment of each of the above mentioned objectives is triggered under certain conditions, i.e., certain conditions of the sensor nodes imply that it requires the mobile device's help in a certain way.

Fault tolerance is given the highest priority because it implies that the sensor node is experiencing a network partition and cannot reach the destination. Thus, the sensor node requires an alternative path to the destination and data must be delivered.

Next energy conservation implies that the sensor node is in dire need of help to reduce its incoming traffic, because if the incoming traffic rate is maintained, the sensor node will soon deplete and will be unable to provide a path towards the destination, thus resulting in network partition.

The next priority is given to congestion reduction, because congestion at the sensor node may result in packet drop due to queue overflow, resulting in lower data delivery to the final destination.

The lowest priority is given to improving end-to-end latency by redirecting traffic

through a faster route. Even if the end-to-end latency is low, the sensor data will eventually get delivered. Therefore, achieving this objective is not the most important goal of the system.

Table 3.1 summarizes the system-defined priority of each of the above mentioned objectives.

Objective	Priority	Synchronize with
Fault Tolerance	Highest	n
Energy Conservation	High	Child(n)
Congestion Reduction	Medium	Child(n)
Latency Reduction	Low	n

Table 3.1: Prioritized objectives to be fulfilled by mobile device along with the action to be taken to achieve each.

3.4.5.3 Actions taken to achieve objectives

When a mobile device selects a sensor node to fulfil any of the above mentioned objectives, the mobile device can take one of the following actions to achieve the given objective for a given node ‘n’ (Table 3.1). For example, if a node ‘n’ is experiencing network partition, then it will be selected by the mobile device for synchronization, however, if the node ‘n’ is low on energy budget, then the mobile device will synchronize with a child of ‘n’ in order to reduce incoming traffic on node ‘n’.

1. **Synchronize with the selected device:** If the network is partitioned, such that any given node is disconnected from the rest of the network and cannot deliver packets to the destination, or if end-to-end latency is to be improved, an alternative path to the destination is to be provided. In these cases, the mobile device synchronizes with the target sensor node to provide this alternative path to the destination.

2. **Synchronize with a child of the selected device:** In order to reduce traffic congestion or energy consumption at any given sensor node, the mobile device must reduce the packets being forwarded by the selected sensor node. In order to reduce packet transmissions by the selected sensor node, the packets being received by the node must be reduced. This can be achieved if the packets from the child node of the selected sensor node are diverted away from the sensor node via the mobile device. Thus, for achieving these objectives, the mobile device synchronizes and communicates with a child node of the selected sensor node.

3.4.5.4 Selection mechanisms to achieve objectives

In order to make conscious choices to select sensor nodes from the set of discovered sensor nodes and take the above mentioned actions to achieve the desired objectives, essentially only two basic selection mechanisms are required representing the actions discussed above, respectively:

1. **Credentials improvement:** This selection mechanism will greedily select a sensor node that has the lowest credentials and forward the data packets from the selected node via the mobile device. Because the selected sensor node had the lowest credentials, i.e., either the sensor node was the farthest away from the destination or not connected at all, the alternative path provided by the mobile device will improve the sensor node's credentials, implying that the connectivity to the destination is established or a number of hops are bypassed.
2. **Traffic reduction:** For each sensor node in the mobile device's vicinity, this selection mechanism selects the child that has transmitted the highest number of packets in the last minute. From this subset of children nodes, the selection mechanism then selects the child node that has the highest number of transmissions in the last minute. However, this selection mechanism may not eventually select a child of the node that is experiencing the highest incoming traffic. This mechanism

is designed so that the highest amount of traffic can be reduced overall in the network, from the nodes that may be a cause of congestion for other sensor nodes.

Due to the nature of the traffic reduction selection mechanism, in selecting a child node with the highest traffic, it may often favour cluster heads due to the higher outgoing traffic on them. Due to this, the cluster heads may experience higher traffic as their credentials improve due to a shorter path offered by the mobile device, thus attracting more traffic from other sensor nodes in the vicinity. However, overall performance of the protocol is expected to improve.

Note that when mobile devices are in the vicinity, higher order nodes (e.g., cluster heads) may forward traffic via lower order node (e.g., end device), thereby inverting the network topology. For this reason, the children of a given node are identified based on its credentials rather than encapsulating the unique address of the parent node in the beacons of the children nodes that can essentially identify absolute children of any given node. The details of the child selection mechanism are discussed in Appendix A.2.

3.4.5.5 Conditions and triggering mechanism for the objectives

As the sensor nodes are expected to broadcast periodic beacons, upon performing a scan of the radio channel, the mobile device will discover a set of sensor nodes, say $DS = \{s1, s2, s3, \dots\}$. Each of the sensor node's beacon will contain performance indicators (Section 3.4.1). As the objectives are prioritized, each objective has to be achieved under different conditions. The performance indicators are used by the mobile device to check whether any of the conditions associated with the prioritized objectives meet. Depending on which condition is met, the mobile device triggers the appropriate objective achievement.

1. **Fault Tolerance:** In the case of network partition, the sensor node will not have a path towards the destination, so its credentials will be close to zero. To keep an error margin, the fault tolerance objective is triggered when the credentials of a given sensor node are below a very small threshold, i.e., $Cr(n) < \tau$. The value

of τ is very small, showing that the sensor node is either disconnected from the network or it is too far from the destination.

2. **Energy Conservation:** If a network partition is not detected, then the next priority is given to the reduction of energy consumption on a sensor node which has low energy reserves, in order to avoid network partition in the near future. Only cluster heads are considered in this objective, because the energy consumption due to transmissions at the source sensor nodes cannot be avoided, as they need to communicate their sensor data anyway (if a connection is available). In addition, if the source sensor node (end device in the tree) is forwarding data on behalf of another sensor node, that traffic will be indistinguishable from the packets originating from the source sensor nodes, by the mobile device without promiscuous listening or an explicit flag in the periodic beacons. Thus, the mobile device will check if the remaining energy of any cluster head is below a certain threshold, i.e., $E(n) < \epsilon$.
3. **Congestion Reduction:** If the conditions for the two objectives with the highest priority are not met, then the mobile device checks whether it can reduce traffic on a critical node. Because a critical node is expected to provide better paths towards the destination, it is expected to exhibit a high traffic density. Thus congestion reduction will be triggered only for those nodes that have transmitted more packets in the last minute than the median of the number of packets transmitted in the last minute by all other sensor nodes discovered by the mobile device.
4. **Latency Reduction:** The accomplishment of latency reduction is triggered when no other objective triggering mechanism has yielded a node for selection.

3.4.5.6 Sensor node selection strategies

As can be seen from the above discussion, different mechanisms suit different objectives, under different conditions. At least four combinations of the triggering and selection

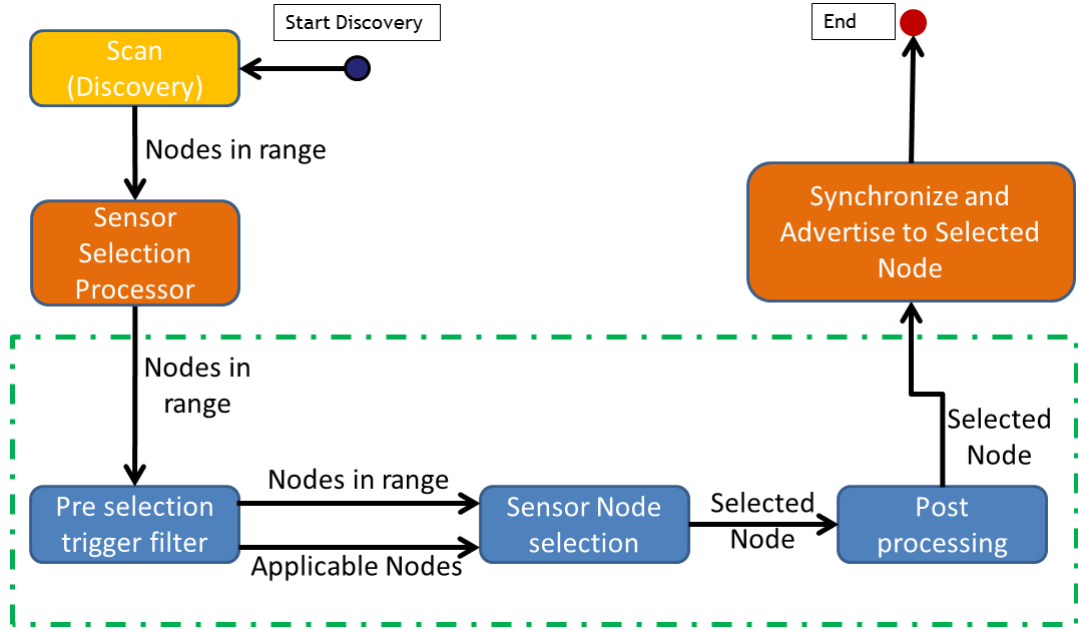


Fig. 3.9: Sensor node selection process at the mobile device.

mechanisms can be identified, based on the accomplishment of the prioritized objectives under different circumstances. Thus the following three stages of sensor node selection by the mobile device are defined:

1. **Pre node selection trigger filter:** This trigger identifies the sensor nodes that will be considered for achieving a given objective.
2. **Actual node selection:** Based on the nodes identified in the trigger filter, this stage will select one of the nodes from the list of nodes in range.
3. **Post processing:** This stage allows any processing required on the selected node.

The sensor node selection process is shown in Figure 3.9, along with the three stages of the process marked inside the dash-dotted box.

All the nodes in range are provided to the “Pre node selection trigger filter”. After short-listing the candidate nodes, this filter provides the list of applicable nodes and

the list of all the nodes in range of the mobile device to the “Node selection” stage. The “Node selection” stage may use all the nodes in range, to identify the children for each of the applicable nodes. After selecting a single node, the “Node selection” stage forwards the selected node to the “Post processing” stage. The “Post processing” stage may choose to modify the selected node in any way that it sees fit.

For each stage, based on the objective to be achieved, different strategies have to be injected. For this reason, the strategy design pattern (Vlissides et al., 1995) is used. In this design pattern, the “Node selection context” is used to hold three different strategies, one for each stage of the process, as opposed to a single strategy defined in the standard pattern.

The “Node selection context” is provided with separate concrete implementations of the strategies to be used in each stage, depending on the objective to be accomplished. The selection process is triggered on the context, which in turn uses the strategies provided from the “Sensor Selection Processor”, shown in Figure 3.9. Both the “Pre node selection trigger filter” and the “Post processing” stages are optional. If any of these stages are to be avoided, no concrete implementation of the strategy for the respective stage needs to be provided.

The strategies provided to the “Node selection context” are listed in Table 3.2 according to the prioritized objective to be accomplished. Each row in Table 3.2 represents the policy applied for the respective objective.

If the “Pre selection trigger” for the policy is not present (marked as NONE in Table 3.2), the set of applicable nodes will be the same as the set of nodes in range. However, if the trigger is present, then the set of applicable nodes will be a subset of the set of nodes in range, i.e., $DS' \subseteq DS$. The set of applicable nodes can be empty as well, depicting that none of the nodes in range fulfil the triggering criteria, and thus the objective will not get triggered, yielding no selected node.

Similarly, if the post processing strategy is not present (marked as NONE in Table 3.2), the selected node (if any) will be returned without any modifications. In the

Objective	Pre selection trigger (Optional) (Section 3.4.5.5)	Node selection mechanism (Mandatory) (Section 3.4.5.4)	Post processing (Optional)
Fault Tolerance	NONE	Credential improvement	Return selected node only if $Cr(n) < \tau$
Energy Conservation	Energy conservation trigger	Traffic reduction	NONE
Congestion Reduction	Congestion reduction trigger	Traffic reduction	NONE
Latency Reduction	NONE	Credential improvement	NONE

Table 3.2: Sensor node selection strategies for prioritized objectives.

case of fault tolerance, if the selected node has credentials higher than τ , then the post processor rejects the selected node, thus no node is selected in this case (Algorithm 1).

For example, a mobile device will trigger accomplishment of energy conservation objective if it finds cluster heads whose energy reserves are below ϵ in the “Pre selection trigger” stage. A list of those cluster heads will then be given to “Traffic reduction” selection mechanism, which will select a child sensor node that is responsible for the highest traffic at the short-listed cluster heads. The selected (child) sensor node will then be given to the “Post processing” strategy. Since, no post processing strategy is available for energy conservation, the selected sensor node will be returned without any modifications back to the “Sensor selection processor” that will synchronize and advertise to the selected sensor node. If no sensor node is selected for energy conservation, i.e.,

either no cluster head was found such that its energy reserves were below ϵ or no child sensor node was found for the short-listed cluster heads, the next policy will be triggered to find a sensor node using which incoming traffic can be reduced at one of the sensor nodes experiencing high traffic.

3.5 Summary

This chapter began by discussing the overall design approach and the challenges involved in designing a protocol that shifts the data forwarding load to more capable mobile devices and reduces the communication load on resource-constrained sensor nodes. The problem was interpreted as a reinforcement learning problem, and four major components, inspired by reinforcement learning, were identified. The alternatives for the placement of decision making component was discussed first. Different aspects of device differentiation and performance evaluation were also discussed, along with long-term and short-term components of a device's delivery ability. Afterwards, challenges associated with neighbourhood discovery were discussed and the approach was identified. In particular, discovery via promiscuous listening versus discovery via periodic beacons was discussed. The section associated with design overview concluded by discussing the use of feedback loop as a mechanism to improve performance of the system. Due to constraints associated with the reactivity of such a feedback mechanism in a highly dynamic environment composed of resource-constrained sensor nodes, only simple MAC layer acknowledgements were used to identify risk of packet loss associated with different types of next hop nodes.

Section 3.3 discussed and analysed the concrete design of the discovery mechanism along with the discovery mechanism available in IEEE802.15.4 (2006) that is used to establish an ad hoc sensor network. In particular, the baseline discovery mechanism employed by the majority of the related work was analysed in terms of listening duty cycle and expected discovery time, where the discovery mechanism is placed at the

resource-constrained sensor nodes. This chapter then discussed the design of the inverse location-aware discovery mechanism in detail, where the discovery mechanism is placed at the more capable mobile device and is only triggered when sensor nodes are expected in the vicinity of the mobile device. The ILAD mechanism was also analysed in terms of listening duty cycle at the resource-constrained sensor node and discovery interval, i.e., the time taken from the arrival of mobile device in the vicinity of the sensor node to the establishment of connection.

Section 3.4 discussed the design of the “STEROID” routing protocol in detail. The structure and architecture of the WSN was discussed first in this section, along with the mechanisms related to network formation. Performance indicators related to device’s delivery ability were also discussed along with their presence in the periodic beacons and advertisement packets of the sensor nodes and mobile devices. In order to simplify the design without compromising decision-making ability, a subset of the components of delivery ability were used as performance indicators.

This chapter also discussed the mechanisms employed in avoiding collisions of periodic beacons within the WSN. A concrete mechanism of credential update was discussed in Section 3.4.2, along with the risk metric used to estimate long term component of device’s delivery ability.

Section 3.4.3 discussed the decision-making component in detail. The decision-making component was sub-divided in to three parts that are partially placed at both sensor nodes (Section 3.4.4) and mobile devices (Section 3.4.5). The three parts of decision making components are responsible for network formation, selection of the next hop neighbour in real-time at the sensor nodes, and intelligently selecting a sensor node to which the services of the mobile device are advertised in accordance with the objectives to be achieved.

The experiences from the EMMON project (Tennina et al., 2011a) in which a large scale WSN was developed, guided the implementation of both the discovery mechanism and the routing protocol discussed in this thesis. In addition to the design, the implemen-

tation itself also has to be simple in order to be implementable on resource-constrained devices in the context of urban-scale WSNs. The TelosB sensor node (Crossbow, 2005) hardware platform was chosen for this thesis as it is one of the most popular and cheapest platforms available, and is widely used in the research community. The TinyOS software stack (Levis et al., 2005; Levis and Gay, 2009) was chosen as it is an open source platform allowing the development of multiple software components using the nesC (Network Embedded System C) programming language without reliance on a typically resource-intensive intermediate virtual machine such as the JVM (Java Virtual Machine) used for example in the more expensive SunSpot (2004) hardware platform that enables software development using abstractions provided in the Java Mobile Edition (J2ME).

As the main implementation hardware platform, TelosB, is very restricted with just 48 kilobytes of ROM storage and just 10 kilobytes of RAM storage available, the implementation of this research was organized in a way to implement only the simplest tasks on the resource-constrained sensor nodes. Incorporating the standard IEEE802.15.4 (2006) MAC layer on such resource-constrained devices is not trivial either due to memory limitations, especially in the presence of additional custom software stacks (such as network and/or application layer) that uses the standard communication protocol in order to perform specialized tasks. Indeed the available implementation of the IEEE-802.15.4 protocol on TinyOS (Levis et al., 2005; Levis and Gay, 2009) alone takes around 38 kilobytes of ROM storage. The space consumed by the IEEE 802.15.4 protocol stack could be even higher depending on the enabled features. This means that the implementation of the resultant network layer protocol, responsible for offloading some communication load to more capable mobile devices, must be light weight in terms of RAM usage on the sensor nodes with a very small code footprint (ROM usage).

To this effect, the sensor node attached to the fixed gateway consumed approximately 39 kilobytes of ROM and 3 kilobytes of RAM, the node attached to the mobile devices consumed approximately 44 kilobytes of ROM and 5 kilobytes of RAM, the cluster head consumed approximately 42 kilobytes of ROM and 3 kilobytes of RAM, and finally

the end node in the cluster-tree consumed approximately 40 kilobytes of ROM and 3 kilobytes of RAM. The usage on mobile device was higher because it was using software components related to serial and radio communication and in addition the mobile device was scanning the radio channel as well.

Chapter 4

Evaluation

This chapter first focuses on the evaluation of the discovery mechanisms (Section 4.1, in terms of actual discovery time and energy consumption at the sensor nodes. The evaluation is conducted in a test-bed environment to ensure accurate energy consumption and operational timing measurements. The test-bed consists of a single sensor node, a single mobile device and a sniffer, to understand the specific effects of the discovery mechanisms in isolation of any other network activity and to achieve results of finer granularity. The evaluation results presented in the subsequent section can be generalized to networks comprising of multiple sensor nodes and mobile devices, as the discovery interval and related energy consumption do not depend on the number of nodes. For the sake of simplicity and repeatability, for these experiments, the mobility of the MDC is emulated using the file based location generator.

This chapter then considers the evaluation of the STERIOD routing protocol that exploits the ILAD discovery mechanism. In order to gain targeted insight into the STERIOD routing protocol, a custom Java-based simulator was developed to evaluate the routing protocol in isolation and quickly establish its viability. In particular, the simulator allowed the execution of multiple combinations of many different parameters in parallel, which would not be feasible in a test-bed environment due to limitations

on hardware availability and contention-free radio channels. Data loss due to mobility of the mobile devices and queue overflow was incorporated in the simulator such that the incoming and outgoing queue lengths were configured to be the same as in the test-bed deployment, and no acknowledgements were generated if the mobile device moved out of the range of the sensor node to which it had advertised (Section 4.2.3). A random waypoint location generator was used to emulate mobile device mobility in the evaluation of the STERIOD protocol to allow many different varieties of the mobility path and to generalize the impact of mobility on the results showing that the protocol works well irrespective of the mobility path. Furthermore, the implementation of the routing protocol in the simulation environment uses common programming paradigms, making it easily portable to a test-bed. This simulation study could be extended by a study in the test-bed environment to evaluate the characteristics of the routing protocol in the real world, and in particular the impact of mobility of the mobile devices and multi-path fading. This will be carried out as part of the future work of this thesis.

Section 4.2 discusses the simulation environment, including the tradeoffs around the simulation of mobility and associated limitations. The evaluation of the “STERIOD” protocol is presented in detail under Section 4.3.

4.1 Discovery Mechanisms Evaluation

This section discusses the evaluation of the discovery mechanisms discussed in Chapter 3 (Section 3.3). Section 4.1.1 presents the experimental design and setup, and Section 4.1.2 discusses the results.

In the conventional discovery mechanism, the sensor nodes scan and the mobile devices broadcast beacons periodically. In the existing alternative discovery mechanism, the sensor nodes broadcast beacons and the mobile devices scan periodically. In the proposed “Inverse Location-Aware Discovery” mechanism, the sensor nodes broadcast beacons periodically, but the mobile devices scan the radio channel only when sensor

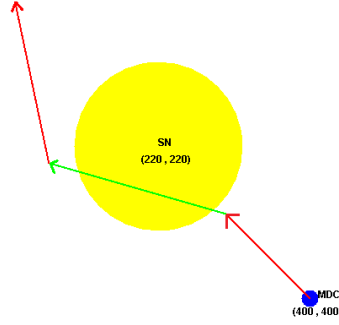


Fig. 4.1: Trajectory of the MDC in a single iteration.

nodes are expected in their range.

4.1.1 Experimental design and setup

Very small networks, each consisting of a sensor node (SN), a mobile data collector (MDC) and a sniffer using TelosB (Crossbow, 2005) sensor nodes, were deployed to evaluate each discovery mechanism.

Mobile devices (MDCs) are assumed to know their location in real-time (e.g., from a GPS). For these experiments, the mobility of the MDC is emulated and the location information is fetched from a configuration file (Section 4.2.4). This configuration file lists a set of initial and final locations and the speed at which the MDC travels between these points. The intermediate locations between the two points are then interpolated by the location provider, every 1.5 seconds. The location information is then encapsulated in all packets being transmitted. The trajectory is shown in Figure 4.1. This trajectory is later extended to incorporate multiple paths.

The receiving devices filter out any packets that they have received from a node located outside their communication radius (set at 100 units), at the network layer, thereby simulating limited communication range. Every node in the network is assumed to know its location, which could be configured on the node itself or could be obtained by a localization algorithm (Chapter 1, Section 1.7.1).

A packet sniffer is used to log all the packets during experiments for later analysis. The system time stamps at which the MDC actually arrives within range of the SN and at which any packet is received at the sniffer, are logged. The time taken to discover a MDC is given as the difference of the system time stamps at which the first advertisement (ADV) packet was sent and when the MDC actually arrived in range of the SN, given by Equation (4.1).

$$T_{D_{actual}} = SysTime_{adv} - SysTime_{InRange} \quad (4.1)$$

Separate networks were established for each discovery mechanism to be evaluated in parallel. Each network ran on its own designated radio channel, eliminating inter-network interference. To avoid any channel-specific characteristics in the results, the radio channels were switched between the networks and the experiments were repeated. Table 4.1 lists the parameters used in experiments. In Table 4.1, the “BackOff” parameter represents the time from the end of a scan cycle to the start of a new one. The “Beacon Order (BO)” parameter dictates the time between two consecutive beacons, i.e., the beacon interval, and the “Superframe Order(SO)” parameter dictates the time period of the active portion of the superframe, as discussed in Chapter 3.

The same trajectory of the MDC was provided from a configuration file (Figure 4.1). The MDC arrives once in range of the SN within a single iteration. Once the MDC reached its destination, it was switched off, essentially stopping all radio communication. The experimental iteration ends at this point. The next iteration begins approximately after 60 seconds, and the whole process is repeated.

The results (Section 4.1.2.1) obtained from these experiments are insufficient, because they only depict a single path. Therefore, further experiments were conducted using the trajectory in Figure 4.2. Using this trajectory, the MDC arrives seven times in range of the SN in a single iteration, via different paths. Each iteration in this case, begins and ends at the same location, i.e., at (400, 400). For this trajectory, multiple combinations of parameters were used (Table 4.2). Again, multiple networks executed on distinct

Experimental Parameter	Value
Beacon Order (BO)	8
Superframe Order (SO)	6
BackOff	30 seconds
IEEE 802.15.4 Radio Channels	25, 26
Transmission Power	-20 dBm

Table 4.1: Common parameters for basic discovery mechanisms, for trajectory in Figure 4.1.

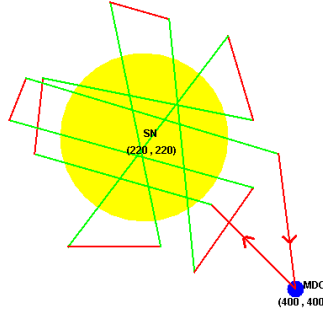


Fig. 4.2: Trajectory of the MDC, arriving 7 times in range in a single iteration.

radio channels in parallel.

Performing any scientific experiment once has no statistical significance, as there could be many variables that are not under the control of the experimental study. Given the intricacy of timing, different runs could have different resultant values. Therefore, high number of runs were needed to approach statistical significance. For this reason, all experiments presented in this thesis were repeated multiple times and the results were continuously monitored. After repeating the experiments using a single path to evaluate two different discovery mechanisms a few times, a singular result of no overlap in the readings was obtained. In order to ensure that the distribution does not change, the experiments were repeated further. As shown in Figure 4.3, the distribution in

Experimental Parameter	Value
Beacon Order (BO)	8
Superframe Order (SO)	6, 4, 2
BackOff	30, 20, 10 seconds
IEEE-802.15.4 Radio Channels	21, 22, 24, 25, 26
Transmission Power	-25 dBm

Table 4.2: Common parameters for all discovery mechanisms, for trajectory in Figure 4.2.

the two cases does not overlap even after repeating the experiments 400 times and no visual difference was monitored, at which point it was concluded that there is a high chance that the distribution depends on the trajectory. In order to remove any impact of the radio channel in use, the channels were switched and then repeated the same number of times. This also had no effect on the distribution of the discovery times for the two mechanisms. This behaviour of having split distribution specially in the case when sensor nodes discover, is thus concluded to be attributed to the trajectory followed by the mobile device. As said previously, having a single path in the trajectory is insufficient. For this reason, the experiments were repeated with multiple trajectories of the mobile device. This resulted in more evenly distributed discovery times, as depicted in Figure 4.5. These experiments were repeated until the battery on sensor node was depleted, resulting in approximately 670 iterations. Achieving the depletion of energy reserves was important to observe the lifetime of sensor nodes for different discovery mechanisms.

4.1.2 Results and Analysis

This section discusses the results obtained from the trajectory in Figure 4.1 in Section 4.1.2.1, and the results obtained from the trajectory in Figure 4.2 in Section 4.1.2.2.

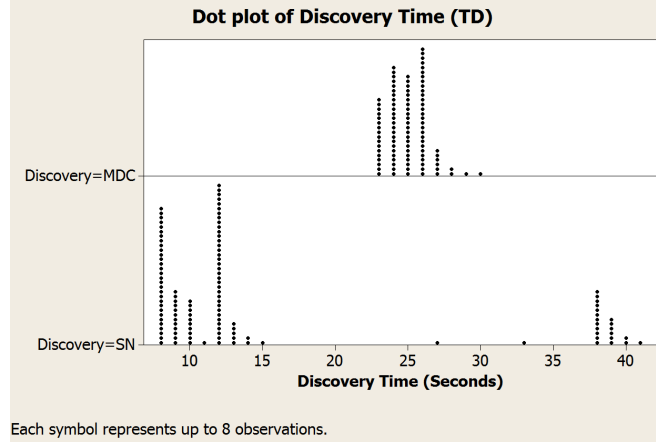


Fig. 4.3: Dotplot of T_D for the existing discovery mechanisms, for the trajectory in Figure 4.1.

Later, Section 4.1.2.3 cumulatively analyses the results.

4.1.2.1 Trajectory with single path

Discovery time:

The dot plot in Figure 4.3 shows the distribution of the actual discovery time ($T_{D_{\text{actual}}}$) when a MDC follows trajectory shown in Figure 4.1.

Repeated experiments show that when the MDC discovers (scans) (existing alternative discovery mechanism), the value of the discovery time (T_D) lies consistently between 22 and 31 seconds. Whereas, when the SN discovers (scans) (conventional discovery mechanism), the value of T_D is between 7 and 16 seconds most of the time, but also quite frequently it is greater than 35 seconds. The higher range is obtained when the MDC arrives in range at the beginning of the BackOff period (Figure 3.7). There are very few occurrences where the T_D is between 16 and 35 seconds when the SN discovers, even though the iterations were repeated more than 800 times in total. No apparent effect of switching the channels between the two implementations was found, as the distribution remained similar irrespective of the channel. This implies that the system

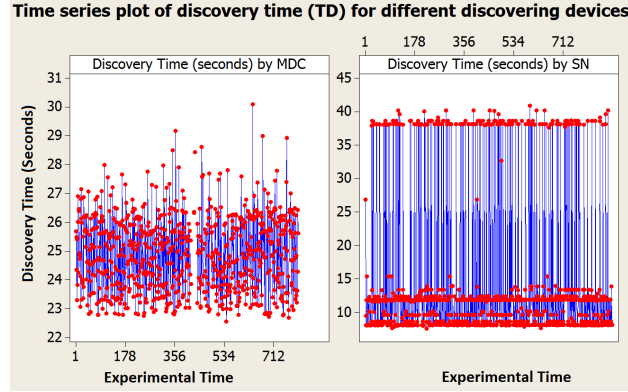


Fig. 4.4: Time series plot of T_D from the existing discovery mechanisms, for the trajectory in Figure 4.1.

discovers MDCs faster when the SN discovers, more frequently though the worst case is small (Table 4.3). The time series plot in Figure 4.4 shows that there is no systematic variation in T_D with respect to time. The main statistical properties of the results are listed in Table 4.3.

The significant difference between the discovery time of the two mechanisms was suspected to be due to the timing of the arrival of MDC in range of the SN, as a result of the trajectory used. Also, the scan duty cycle is interlinked with when the MDC was switched on in the case when MDC discovers. However, when the SN discovers, the scan duty cycle is more independent of the mobility and switching on of the MDC. In order to obtain a more accurate picture, the experiments were generalized to use more than one paths.

4.1.2.2 Trajectory with multiple paths

Discovery time:

In order to obtain a generalized perspective, the experiments were repeated with the trajectory given in Figure 4.2. As expected, these generalized results exhibit a different pattern. The dot plot in Figure 4.5 shows that values of T_D are more evenly spread

Discovering Device	MDC	SN
$T_{D_{min}}$ (seconds)	22.56	7.59
$T_{D_{max}}$ (seconds)	30.09	40.90
$T_{D_{average}}$ (seconds)	24.90	15.24
Standard Deviation of T_D	1.29	10.94

Table 4.3: Observed properties from experiments, for the trajectory in Figure 4.1.

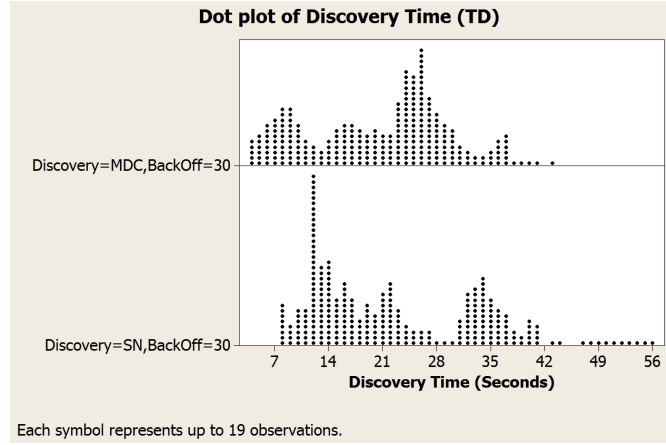


Fig. 4.5: Dotplot of T_D for the basic discovery mechanisms, following the trajectory in Figure 4.2.

for both existing discovery mechanisms, while using this extended trajectory for the same setting of BackOff and Beacon Order (BO) parameters from Table 4.1. In fact, T_D 's distribution seems to be shifted a little towards the higher values when the SN discovers. From Figure 4.5, it seems like the MDC is discovered faster more frequently when the SN scans. Overall, however, the dot plot seems to validate that the distribution becomes more smooth as the paths followed by the MDC are increased, when compared to Figure 4.3.

It is expected that increasing the number of different paths followed by the MDC, converges the estimate of $T_{D_{average}}$ towards its true value, for both existing discovery

mechanisms. Statistical observations from the extended trajectory (Figure 4.2), are provided in Table 4.4. The value of $T_{D_{average}}$ has significantly increased from 15.24 to 21.74 when the SN discovers, compared to the previously observed value for the same values of *BackOff* (30 seconds) and BO (8). However, $T_{D_{average}}$ decreased from 24.90 to 20.106 when the MDC discovers for the same *BackOff* and BO (Tables 4.3 and 4.4).

Discovering Device	MDC				SN		
BackOff (seconds)	30	20	10	ILAD	30	20	10
$T_{D_{min}}$ (seconds)	3.79	3.79	3.77	5.67	7.53	7.88	7.54
$T_{D_{max}}$ (seconds)	43.33	35.48	18.77	13.17	55.79	45.84	35.78
$T_{D_{average}}$ (seconds)	20.11	15.63	10.94	10.46	21.74	19.39	15.16
$T_{D_{median}}$ (seconds)	22.73	15.36	10.09	10.43	19.14	16.58	15.24
Standard Deviation of T_D	8.76	4.65	4.58	1.29	10.11	6.46	4.44
95% CI of T_D	(19.91,20.3)	(15.43,15.82)	(10.75,11.13)	(10.02,10.89)	(21.55,21.93)	(19.20,19.59)	(14.97,15.35)
99% of T_D (seconds)	< 37.10	< 28.89	< 18.59	13	< 41.22	< 39.36	< 25.38
% of T_D under expected limit	99.89	99.93	100	100	96.48	93.71	85.85

Table 4.4: Observed properties from experiments, for trajectory in Figure 4.2, BO=8.

Table 4.4 also shows that 99% of the T_D observations are less than 37.10 seconds when the MDC discovers and are less than 41.22 seconds when the SN discovers, for the same BackOff value of 30 seconds. An overwhelming majority of the T_D observations lies within the expected interval. However, in some cases (less than 15%), T_D is higher than the expected interval. Typically, this was observed to be due to the interference from external sources such as WiFi. If due to interference the ADV packet is corrupted, successful discovery is unidentifiable. Therefore, the next ADV packet is used to calculate $T_{D_{actual}}$, increasing the observed discovery time (T_D). Also missed beacons (due to temporary software fault) by the scanning device impact the discovery time, as the scanning device will only successfully discover in the next scan cycle.

The dot plot in Figure 4.6 is obtained by keeping BO constant at 8, and reducing the BackOff according to Table 4.2. It shows that reducing the value of BackOff has no effect on $T_{D_{min}}$, but reduces $T_{D_{max}}$, hence reducing the interval of T_D observations for the same discovery mechanism, as expected. In general, $T_{D_{min}}$ and $T_{D_{max}}$ are observed to be lower

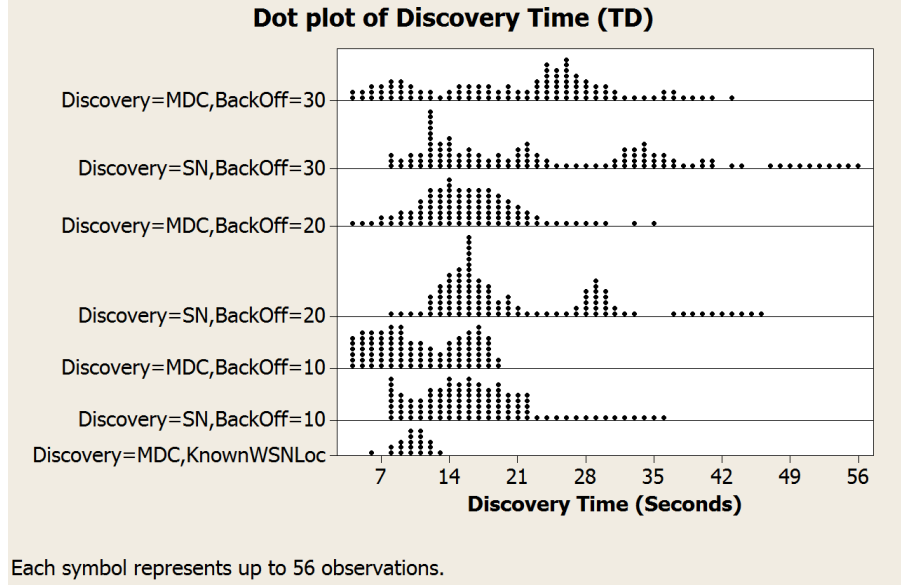


Fig. 4.6: Comparative dotplot of T_D for all discovery mechanisms, following trajectory in Figure 4.2 and BackOff in Table 4.2.

when the MDC discovers a beaconing SN, than when a SN discovers beaconing MDC for the same parameters. In addition, it shows that the distribution of the discovery time becomes smoother when reducing the BackOff value for both existing discovery mechanisms (Figure 4.6)¹.

The discovery interval can be reduced by reducing the BackOff value and keeping the value of BO constant. In addition, lower interval bounds can be achieved when the MDC discovers, for the same BackOff and BO parameters (Table 4.4).

Figure 4.6, also shows that the ILAD mechanism discussed in Section 3.3.4 of Chapter 3, where the location of the sensor network is known by the mobile device, performs much better than the two existing mechanisms. In this case, the discovery interval is much smaller because the mobile device only starts scanning when it expects a wireless sensor network (sensor node in this case). This result confirms the hypothesis that a

¹Note that, the distribution of the two existing discovery mechanisms being very similar, it might be that as the number of trajectories increases, the distribution becomes identical

lower discovery interval can be achieved by using the location information of the sensor network.

Energy:

Apart from the discovery time, the energy consumption at the SN for both basic discovery mechanisms with different parameters (Table 4.2) was also analyzed, using the trajectory in Figure 4.2. This analysis was only necessary for the SN because this work aims to reduce the energy consumption only on such devices (Chapter 1, Section 1.7.1). For this analysis, two new AA batteries were used to power each SN. In order to obtain the current values of the energy reserves, the on-board voltage sensor of the TelosB (Crossbow, 2005) SN was used. The voltage readings reflect the remaining energy on batteries of each SN. The voltage reading is obtained in its raw form, and the actual voltage can be calculated by Equation (4.2), where $V_{ref} = 1.5$ V, representing the voltage provided by a single AA battery, and is therefore multiplied by 2. The maximum voltage on a SN could be 3 volts. As the energy reserves deplete, the output voltage of the battery reduces as well. Once the output voltage of the battery crosses the lower threshold of 2.1 volts, the CC2420 radio (Chipcon, 2006) ceases to function.

$$Voltage = \frac{VoltageSensorReading}{4096} * V_{ref} * 2 \quad (4.2)$$

The raw voltage sensor reading is encapsulated in packets, and is thus captured in the sniffer log. The voltage is then plotted against time for different parameters (Table 4.2). The curves obtained from these plots, represent the battery drain. Steeper curves show more energy consumption than their counterpart.

Figure 4.7 instigates the effect of varying the duty cycle. This is done by changing the values of BackOff (time between two scan cycles) when the SN scans periodically, and the values of superframe order (SO) (active portion of the superframe) when the MDC scans periodically. The BackOff and SO parameters are used from Table 4.2. When the SN discovers the MDC by scanning periodically, the energy curves decline much

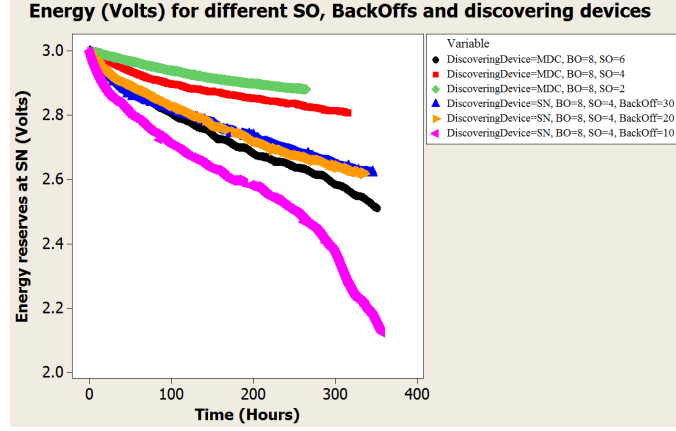


Fig. 4.7: Energy consumption at SN vs. time for different SO, BackOff and discovery mechanisms, for the trajectory in Figure 4.2.

faster with decreasing values of BackOff. This phenomenon confirms that by reducing the BackOff value, the listening duty cycle increases because the scanning is done more frequently, thus increasing the energy consumption.

Figure 4.7 also shows that when the MDC scans periodically in order to discover the SN, the energy consumption at the SN increases with the increasing value of the SO parameter, which dictates the portion of the superframe during which the radio at SN is actively listening. Thus, for a constant BO, increasing the value of SO increases the listening duty cycle, resulting in a faster decline in energy reserves at the SN. An interesting observation is that the energy curves are similar when BackOff is set at 30 and 20 (when the SN discovers) and when SO is set at 6 (when the MDC discovers). This behaviour is observed because the listening duty cycles for these settings are similar, as estimated by Equations (3.7) and (3.12). In addition, when the MDC discovers, if the value of SO is lower than 6, the energy consumption at the SN becomes much smaller compared to the discovery mechanism where the SN discovers, because the duty cycle decreases significantly.

Figure 4.8 shows the voltage curves for different values of BackOff while keeping BO

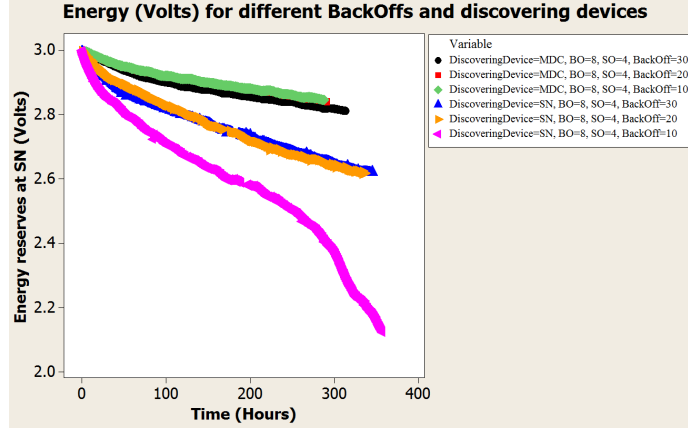


Fig. 4.8: Energy consumption at SN vs. time for different BackOff and discovery mechanisms, for the trajectory in Figure 4.2.

and SO constant at 8 and 4 respectively, for both basic discovery mechanisms. When the MDC discovers, changing the value of BackOff has no significant effect on the energy consumption at the SN, however it is significantly more efficient than the consumption by the conventional discovery mechanism.

Note that the discussion and the results of the energy consumption on the sensor nodes are the same for the ILAD mechanism (Chapter 3, Section 3.3.4), where the mobile devices (MDCs) scan the radio channel when they expect sensor nodes in their vicinity.

4.1.2.3 Discussion

These results show that (at least) for the given trajectory (Figure 4.2), the proposed discovery mechanisms (Chapter 3, Sections 3.3.3 and 3.3.4) perform much better than the conventional discovery mechanism (Chapter 3, Section 3.3.2) in terms of both discovery time and energy consumption. However, trajectories comprising many different paths are expected to show similar results.

In the alternative discovery mechanism discussed in Chapter 3, Section 3.3.3, the discovery time can be decreased by reducing the BackOff parameter that is only ap-

plicable at the MDC without significant impact on the energy consumption at the SN, thus reducing the interval of the discovery time. For this mechanism, however, the SO parameter should not be reduced too much, as it will impact the actual data communication time, as it dictates the time period during which a sensor node will actively listen for data communication.

If the application requirement in terms of discovery time is not stringent, i.e., the MDC is assumed to move at lower speeds, the conventional discovery mechanism (Chapter 3, Section 3.3.2) may perform better in terms of energy consumption due to a lower listening duty cycle while having a large value of BackOff. In addition, certain classes of applications such as target tracking applications may be better served with the conventional discovery mechanism, because the target could be hostile and the SN would have to identify the target's presence without beaconing.

In essence, if the required maximum discovery time and minimum communication time are known (from the application/network requirements), the appropriate discovery mechanism can be chosen, by trading off energy consumption. In addition, the alternative discovery mechanism discussed in Chapter 3 (Section 3.3.3), being more deterministic, can provide better control of the required timing. Intelligently adaptive algorithms may also perform better when applied to the alternative discovery mechanism.

Furthermore, since MDCs know their location, and the SNs are assumed to be fixed, the MDC can learn the WSN location from experience and thus trigger discovery mechanism only when inside a SN's vicinity. The results show that knowing the location of the sensor nodes significantly reduces the discovery interval. Thus the "Inverse Location Aware Discovery" mechanism (Chapter 3, Section 3.3.4) can discover the sensor nodes much quicker. As the discovery is performed at the mobile device, the sensor nodes can be configured at certain duty cycle using BO and SO parameters, making their energy consumption deterministic. In this case, the accuracy of the localization mechanism will have almost no impact on the energy consumption of the sensor nodes.

4.2 Simulation

In order to quickly evaluate the STERIOD routing protocol in a number of different deployment scenarios with a different number of mobile devices moving in the vicinity of the sensor network with varying speeds, a custom Java-based simulator was developed. The deployment scenarios were loaded from a configuration file that represented different network deployments, in the simulation. The deployment scenarios are discussed later under Section 4.3.2.

As discussed in the introduction of this chapter, the simulated environment allowed the experiments to be performed efficiently with reproducible results, while allowing individual characteristics to be studied in isolation. The implementation hierarchy of different node types available in the environment are discussed in Appendix D.1.

The same mechanisms were used for network formation, next hop selection at the sensor nodes, and sensor node selection at the mobile devices, as in the actual implementation in the test-bed environment. However, the test-bed was only used to verify and validate that the routing protocol can actually be implemented on the sensor nodes and the protocol is workable in a real environment, whereas, the simulator was used for the performance evaluation of the protocol. The main difference between simulation and test-bed was the communication layer.

The communication itself was simulated. In order to simulate the sensor nodes, fixed length transmission and reception queues were used. This means that at any time, only a fixed number of packets could be stored in the transmission and reception queues of the sensor nodes. If a queue was full, the packets were dropped from that queue. This allowed the simulator to emulate the memory-constrained nature of the sensor nodes.

As shown in Figure 4.9, the data loss due to the mobility of the mobile devices is also simulated. In particular, the mobile device does not generate an acknowledgement (of received data packet) if it has moved out of the range of the selected sensor node. In this case, a data packet sent via the mobile device is redirected towards the parent

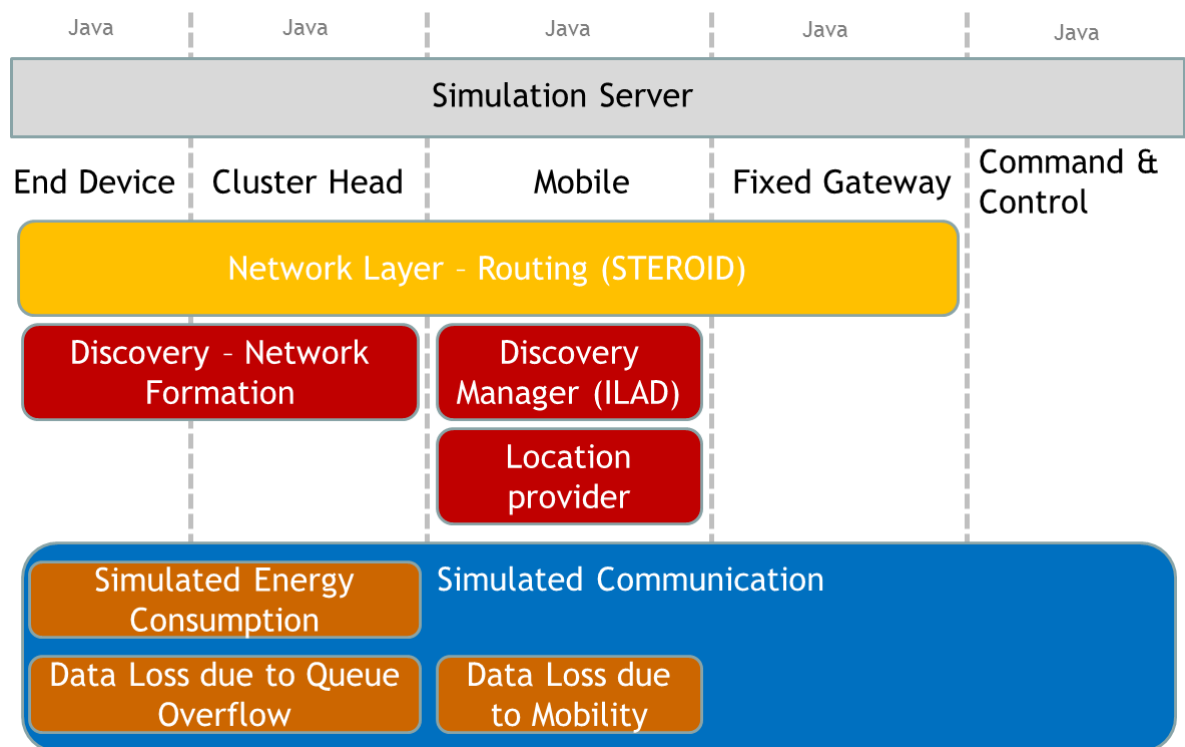


Fig. 4.9: Overview of the components developed for implementing Inverse Location-Aware Discovery mechanism and STEROID protocol in simulated environment.

node, in order to reduce data loss, and the risk metric is updated at the source node.

In addition, the simulator also emulates the energy consumption related to communication. The energy consumption is based on the specifications of the CC2420 (Chipcon, 2006) radio module. The emulation of energy consumption due to communication is discussed later in detail in Section 4.2.5.

The remainder of this section first discusses the discovery manager that is implemented at the mobile devices and is responsible for discovering sensor nodes using the ILAD mechanism in Section 4.2.1. Section 4.2.2 discusses how networks are deployed under the simulated environment. Sections 4.2.3 and 4.2.4 discuss how packet loss on the sensor nodes and the mobility of mobile devices are simulated, respectively. Section 4.2.4 also discusses the advantages and disadvantages of using random mobility in the simulated environment, particularly in comparison with real world mobility traces. Section 4.2.5 discusses the simulation of energy consumption within the context of the simulated environment and how energy dissipation is simulated on the sensor nodes. Finally, this section ends with a discussion on the limitations of the simulated environment used in this thesis in Section 4.2.6.

4.2.1 Discovery manager

The discovery manager is responsible for initiating radio channel scans on the mobile device and selecting a suitable sensor node in its vicinity to offer its services for offloading some communication load away from the resource-constrained sensor nodes.

The location of the sensor nodes is fetched from the command and control server, and the location of the mobile device is updated regularly using the location provider (Section 4.2.4, Figure 4.11). The discovery manager uses this location information to perform Inverse location-aware discovery (Chapter 3, Section 3.3.4). The discovery manager initiates radio channel scans when it expects sensor nodes in its vicinity.

Once the radio channel scan is initiated, the beacons received during the scan duration are forwarded to the discovery manager. The discovery manager stores the received

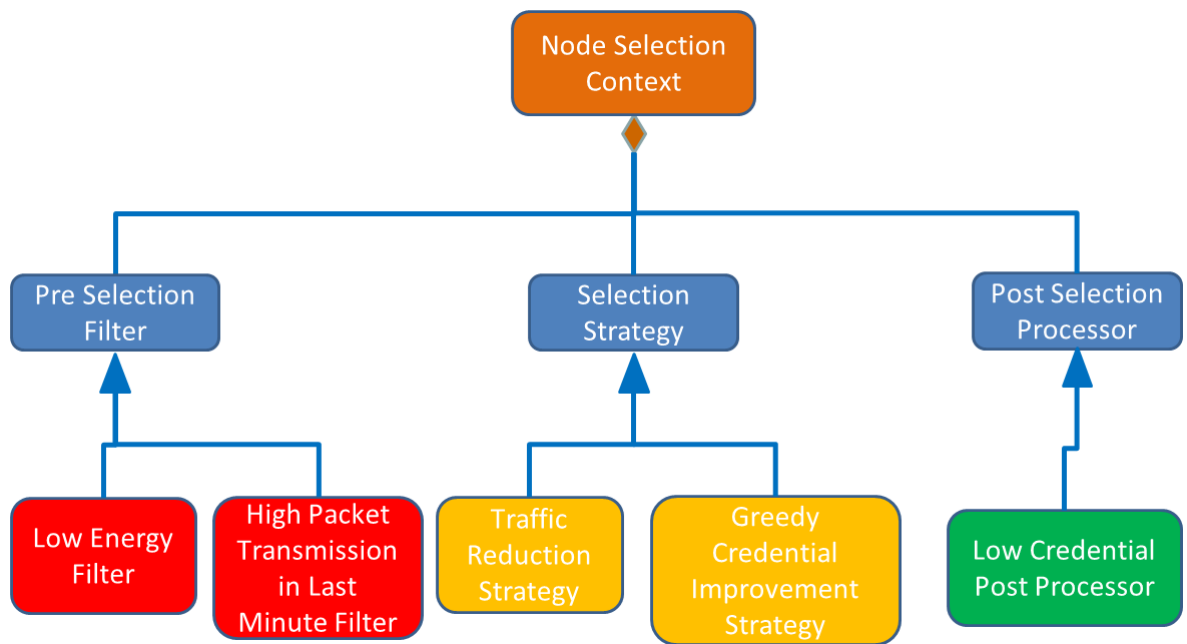


Fig. 4.10: Variant of the strategy pattern, implemented to perform sensor node selection by the mobile device using different combinations of pre selection filter, selection strategy, and post processor, in order to achieve prioritized objectives.

beacon packets locally, to be used to select a sensor node in its vicinity. Upon completion of the scan, the discovery manager initiates its selection process as described in Chapter 3, Section 3.4.5.

The sensor node selection process is implemented as a variant of the strategy design pattern (Vlissides et al., 1995), such that three separate strategies are used for each stage of the selection process. As shown in Figure 4.10, the Node Selection Context is composed of a Pre-selection filter, a Selection strategy and a Post-selection processor. The concrete implementations of each stage depend on the prioritized objective to be met according to Table 3.2 in Chapter 3. The details of the selection process are discussed in Chapter 3, Section 3.4.5

The node selection process is repeated for each prioritized objective at the mobile device. Once a node is selected, the discovery manager at the host mobile device requests synchronization with the selected sensor node. When the synchronization is complete, the discovery manager sends an advertisement (ADV) packet to the selected sensor node. This allows the selected sensor node to become aware of the presence of a mobile device and in turn allows the selected sensor node to forward data packets towards the command and control server via the mobile device.

4.2.2 Network deployment scenarios

For each experiment, the network deployment was imported from an external file in the simulator. The network deployment file consisted of the location of each individual node with the exception of the mobile devices and the command and control server. The command and control server was assumed to be located in a distant data warehouse behind an IP network (cloud).

The location of the mobile devices was generated using a “Random WayPoint Generator” that generated the location of a mobile device based on the mobility traces resulting from the random waypoint mobility model, with 0 pause time (simulating continuous motion) and random speed. The maximum speed was configured in the simulator per

experiment. The maximum speed was configured at 10 units/second for all experimental configurations except for the experiments where the effect of speed of mobile devices was evaluated, whereas, the transmission range was set at 100 units (Section 4.1.1). This speed allowed the simulation of a slow moving vehicle, e.g., when the vehicle approach a junction.

The experiments were set up in different network deployment scenarios. Initially the networks were planned to study the effect of the location of the source nodes and their connectivity on other factors. Later, the network deployment was semi-planned (generated algorithmically) to generalize the observations.

In each experimental run, an increasing number of mobile devices were used in order to study the impact of the number of mobile devices on different aspects of the network, as explained in detail later in this chapter.

In these experiments, only a subset of the deployed sensor nodes actually generated the packets. These source sensor nodes are assumed to be the edge of the network tree (in terms of connectivity), as the intermediary nodes (cluster heads) are assumed to be only responsible for data forwarding. This means that these source sensor nodes do not become parent of any other sensor node. However, if a mobile device is connected to these source nodes, then the source nodes do offer shorter paths to their parents in a hierarchical tree network topology.

The source nodes generated a packet every 15 seconds, simulating a low throughput network. All sensor nodes broadcasted a periodic beacon every 4 seconds (approximately simulating a BO value of 8 in the IEEE802.15.4 (2006) standard). This beacon interval was selected based on the experiments carried out in the evaluation of the ILAD mechanism as the configuration showed positive results (Section 4.1.2).

Each simulation run was executed for 50 minutes, allowing heap memory refresh every hour along with a 10 minutes resource clean up window for the operating system. The results from the experiment runs were continuously monitored. The experiments were stopped once the error bars across multiple graphs ceased to be visible. This resulted in

repeating each simulation run approximately 35 times.

4.2.3 Simulating Packet Loss

This simulator implemented packet loss due to mobility by discarding the packets that were received at the mobile node after the mobile node moved out of the transmission range of the stationary node. In this case, no acknowledgement was issued. By detecting the missing acknowledgement the resource-constrained sensor nodes were able to identify the connection loss. This strategy mimics the identification of connection loss in a real environment.

In addition, the simulator implemented congestion due to overflowing packet queues, by implementing queues of limited size at the resource-constrained sensor nodes. Any packets received after the packet queue was filled were simply discarded. In this case, an acknowledgement was issued so the previous hop was under the impression that the packet was successfully forwarded, however, in reality the packet was dropped due to queue overflow. This was also implemented in this way to closely simulate the actual implementation.

4.2.4 Simulating Mobility

The command and control server is assumed to be located in a remote data center, and thus is not considered to be mobile. The fixed gateways are assumed to be stationary devices allowing consistent network access to the WSN and are deployed in its vicinity. Therefore, the location provider component is only required for mobile devices, as these are the only devices that are considered to move in the vicinity of the WSN. The location provider is assumed to provide location in a two dimensional cartesian coordinate system.

Figure 4.11 shows the implementation hierarchy of the location provider. The location provider interface provides a simple access point for the mobile device from where it can acquire its current location. In this research, the mobility pattern is emulated either by File based generator or Random Waypoint generator.

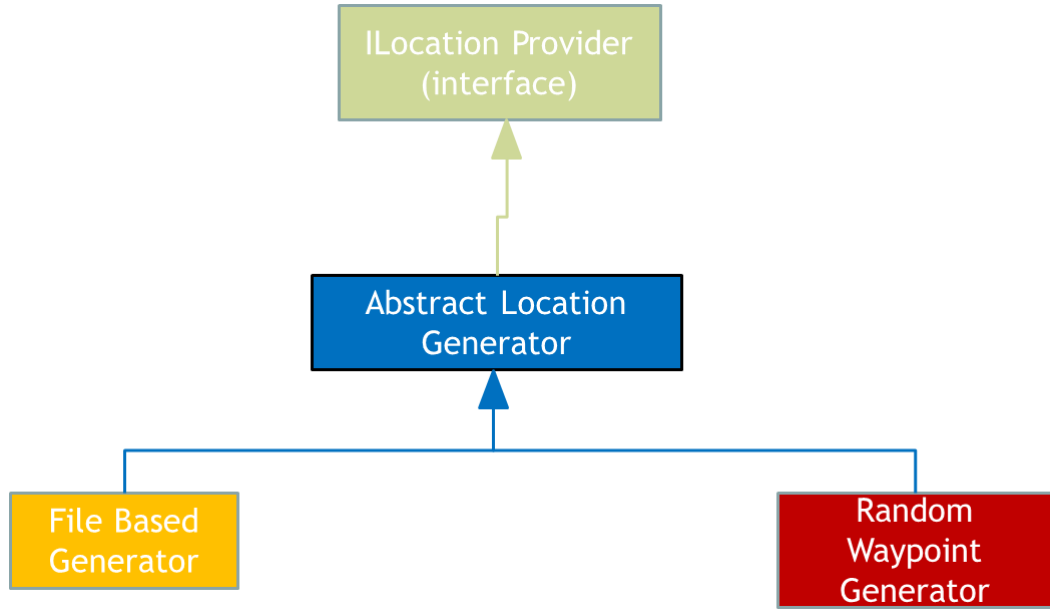


Fig. 4.11: Location generators emulating mobility on mobile devices.

The file-based generator provides the mobility pattern defined in a configuration file. This allows the mobile devices to follow a fixed path and is useful to carry out repeatable experiments using a fixed trajectory. The file-based generator provides the initial and final locations for each stride along with the speed at which the mobile device will move on the given stride, from the configuration file. Once the mobile device reaches the final location of the stride, the file-based generator loads the next stride from the configuration file, which starts from this final location. The file-based location generator was used in the experiments carried out to evaluate the ILAD mechanism (Section 4.1) to allow repeatability.

The random waypoint generator also provides the initial and final locations for each stride along with the speed at which the mobile device will move during the stride. However, the initial and final locations and the speed are randomly generated. The random waypoint generator assumes zero (0) pause time. Once the final location is crossed, the next destination is randomly selected with a new random speed. The random

waypoint location generator was used to evaluate the STERIOD protocol (Section 4.3) to allow mobility path variety.

Both these concrete generators inherit from an abstract location generator class that provides common functionality to both. The abstract location generator calculates the intermediary locations based on the direction inferred from the initial and final locations and the speed given by the concrete location generators. The abstract location generator only requests a new destination and speed from the concrete generator once the final location is passed. Each time that the network controller requests the current location, the abstract location generator interpolates the location based on the time passed since the last request and the velocity (direction and speed) of the mobile device. For interpolating the current location, trigonometric functions are used (Appendix D.2).

In the real-world the current location can be fetched from an external source such as GPS. In that case, only the implementation of the concrete location generator will be replaced while keeping rest of the implementation agnostic to the actual source of the current location.

4.2.4.1 Random mobility versus real scenario

As discussed in the chapter introduction fixed mobility patterns were used to evaluate the ILAD discovery mechanism and random mobility patterns were used to evaluate the STERIOD routing protocol.

In a real mobility scenario, individuals and vehicles may very often move along the paths and roads that are laid out around an urban town. The mobility of individuals and vehicles is not random in such a case. The participating individuals and vehicles may come in contact with sensors at certain instants for a certain time periods, for example when the traffic lights turn red. This mobility behaviour results in bursty communications between sensor nodes and mobile devices, such that sensor nodes will route packets via mobile devices during these contact intervals. Such a behaviour is not thoroughly studied in this research, however, this research can be extended in the future

to incorporate the bursty nature of contacts. In such scenario, the sensor nodes could store some packets until they come in contact with mobile devices and then route via mobile devices in bursts of packets.

In such cases, the sensors may not be able to route their data via participating mobile devices all the time. When mobile devices are not available, baseline performance will be achieved in the STERIOD routing protocol as it does not completely rely on mobile devices rather opportunistically uses them when available. The following sections show the performance of the routing protocol with zero mobile devices in each study. However, there is no performance hit on the sensor nodes due to superfluous discovery of mobile devices when they are not present as sensors do not pro-actively try to find mobile devices, rather the mobile devices advertise themselves when they are in the vicinity of the sensor nodes due to ILAD discovery mechanism.

4.2.5 Simulating Energy Consumption

Modelling the energy consumption in WSNs is an open domain of research in its own right (Dunkels et al., 2007; Ahmed et al., 2009). As the modelling of energy consumption itself is out of the scope of this research, a simple technique to model the energy consumption was adapted for the purposes of comparing different scenarios.

As current is defined as the rate of flow of electric charge, the amount of the electric charge consumed for each operation can be calculated as $C = I \times T$, i.e., as the product of the current consumption of the operation by the time duration of the operation. In addition, as the capacity of batteries is also calibrated in the same unit, the consumption of electric charge can be evaluated simply, if assuming a linear energy dissipation model (note that this assumption may not hold, but is used for simplicity). In addition, the computation can also be simplified by assuming that the battery provides a constant voltage output to the connected circuit as the electric charge on the battery dissipates. Note that this second assumption is also not close to reality, as seen in Section 4.1.2, the output voltage drops as the energy dissipates on the battery. The CC2420 radio chip

ceases its operation as the output voltage provided by the battery drops below 2.1 volts. However, this assumption can be used as the energy consumption is only required for comparison within the same simulated environment.

In order to simulate the energy consumption, the specifications of the CC2420 radio module (Chipcon, 2006) were used. The parameters for reception, transmission and idle mode are chosen to be the same as the real environments, and the corresponding rate of current consumption is used. The current consumption during reception, transmission and idle mode were configured as 19.7 mA, 8.5 mA and 20 μ A respectively, while ignoring the current consumption during processing and sensing operations, where mA denotes the unit of current consumption as milliamperes and μ A reflects the unit as microamperes. The current consumption during processing and sensing operations was ignored as it was assumed to be negligible compared to transmission and reception operations (Chipcon, 2006). In addition, the processing and sensing operations were not of interest for this research.

Thus, reusing the online componentized energy estimation model of Dunkels et al. (2007), the simulation was configured with the maximum capacity of an imaginary battery set at 1 milliamperes hour, meaning that if 1 milli ampere is consumed from the battery, then the battery will last for a single hour. Thus energy consumption was calculated according to Equation (4.3), where I_c and T_c represent the amount of current consumed (in amperes) by the operation and the duration of the operation (in seconds), respectively. Initially, the value of C_t represents the remaining electric charge and was configured at the value of the battery capacity.

$$C_{t+1} = C_t - I_c \times T_c \quad (4.3)$$

Each time the radio was activated, the amount of current consumed during the respective operation was reduced from the maximum capacity of the battery, providing us with the remaining battery capacity. Even though, in reality, the order of battery capacity is much higher, of the order of approximately 1200 mAH (depending on the

manufacturing process), the battery capacity was kept very low (1 mAH) to highlight node death.

When a sensor node transmits its beacon, it is assumed to have consumed an electric charge for a single idle time, a single superframe duration in the listening mode, and a single beacon in the transmission mode. When a sensor node receives a beacon, it is assumed to have consumed an electric charge for a single beacon and a single data packet in the listening mode. When a sensor node transmits a data or advertisement packet, it is assumed to have consumed an electric charge in the transmission mode according to the packet size.

For the experiments, the values of the beacon order and superframe order were assumed to be configured as 8 and 4 respectively, as these values resulted in an efficient listening duty cycle at the sensor nodes, as discussed in the evaluation of the discovery mechanisms (Section 4.1). Using Equations (3.2) and (3.3), and assuming that one base superframe duration approximately equals to 0.01536 seconds (Lu et al., 2004), the beacon interval and superframe durations were calculated.

For the experimentation, the idle time T_{idle} was derived from Equation (4.4) where BI and T_{sfd} were derived from Equations (3.2) and (3.3), respectively. Thus, the charge consumed during a single idle time was simply calculated as $I_{idle} \times T_{idle}$.

$$T_{idle} = BI - T_{sfd} \quad (4.4)$$

The time taken to transmit a packet was calculated by Equation (4.5) in seconds. From the CC2420 specifications (Chipcon, 2006), the transmission rate is given as 250 Kbps. For simulating the correct behaviour, the units were converted accordingly into bits per second. The sizes of the packets transmitted by the sensor nodes are given in Table 4.5.

$$T_c = \frac{PacketSize}{RadioTransmissionRate} \quad (4.5)$$

Packet Type	Packet Size
Data	36 bytes
Advertisement	25 bytes
Beacon	23 bytes

Table 4.5: Packet sizes for different packet types used in the simulation.

4.2.6 Limitations of the simulator

Having a simulated environment allowed for the quick validation of the work presented in this thesis. The motivation for using a simulator were discussed in the beginning of this chapter.

The simulator, however, does not simulate multi-path channel fading and radio channel interference due to third party devices that emit electromagnetic radiation. Such devices include 802.11 (WiFi) and 802.15.1 (Bluetooth) traffic in the vicinity of the sensor network. In addition, harmonics from other radio communication such as GSM (cellular) communication and FM/AM radio broadcasts may also affect the communication carried out by the sensor network.

Furthermore, the simulator did not simulate links with bad quality, i.e., all links are assumed to be stable for the purposes of evaluation of the STERIOD protocol. The instability of the links only arose due to mobility of the mobile devices in this simulator.

In addition, the effect of realistic mobility pattern has not been studied in this research, which could potentially impact the performance of the ILAD discovery mechanism. Because only random mobility was emulated when evaluating STERIOD routing protocol, the effect of non-random mobility pattern was not studied. This is previously also discussed in Section 4.2.4.1.

The simulator implemented in this research does not simulate real energy consumption, rather only simulates linear energy consumption pattern for comparison purposes only. Thus, a real implementation would also provide detailed analysis on the energy

consumption of the sensor nodes, as discussed in Sections 4.2.5 and 4.1.2. The energy consumption on smart mobile devices was not simulated either as it was not of interest for this research, while the energy comparison was only carried out on the sensor nodes.

The simulator served its purpose in this research, a complete study, however, would require the full protocol to be evaluated in a real environment, where sensor nodes are deployed in an urban town center and smart mobile devices are used to offload communication load away from them. Such an environment requires significant resources and expertise which remain out of the scope of this thesis.

4.3 STERIOD Routing Protocol Evaluation

The purpose of the STERIOD routing protocol is to reduce the communication load at resource-constrained sensor nodes by utilizing mobile devices that are abundantly present in urban environments. Reducing the load on sensor nodes is expected to result in overall better performance in terms of end-to-end latency, congestion and energy consumption. At the same time routing via paths through mobile devices means a potential increase in the risk of data loss.

This section aims to study the effect of the STERIOD protocol on the sensor network. Section 4.3.1 outlines the objectives for the evaluation. Section 4.3.2 discusses different network configurations used for the evaluation of the STERIOD protocol. This section then evaluates a few design choices in Sections 4.3.3 and 4.3.4. Section 4.3.3 discusses the impact of mobility on data deliverability and the technique used to mitigate data loss in the STERIOD protocol. Section 4.3.4 discusses the effect of the selection mechanisms used by the mobile device for selecting a sensor node. Sections 4.3.5 to 4.3.7 discuss the effect of employing the STERIOD protocol in terms of end-to-end latency, communication load and fault tolerance, respectively. Section 4.3.8 discusses the effect of employing the STERIOD protocol in terms of energy consumption and Section 4.3.9 discusses energy consumption on the mobile devices depending on the usage of different

radio communication modes. Section 4.3.10 discusses the impact of varying speed on the STERIOD protocol. Section 4.3.11 performs a comparative study of the STERIOD protocol against the DELAR (Liu et al., 2011) protocol that is designed to redirect traffic towards more capable nodes in the network. Finally Section 4.3.12 concludes this section.

4.3.1 Objectives of the Evaluation

This chapter, in Section 4.3, studies the network characteristics after implementing the STERIOD protocol within the wireless sensor network (WSN) domain. The studied characteristics are based on the goals and requirements outlined in Chapter 1 under Section 1.5, and are as follows:

1. Data loss while forwarding via mobile devices due to their mobility.
2. End-to-end latency while delivering packets to the destination.
3. Congestion at the sensor nodes.
4. Load on sensor nodes.
5. Fault tolerance in case of network partitions (by availing mobile devices).
6. Energy consumption at the intermediary sensor nodes.
7. In addition, a comparative study with the DELAR protocol (Liu et al., 2011), as it was identified as the most suitable baseline in Chapter 2.

It is expected that by reducing the traffic within the sensor network, the protocol will result in:

1. Reduced end-to-end delays.
2. Improved energy conservation.

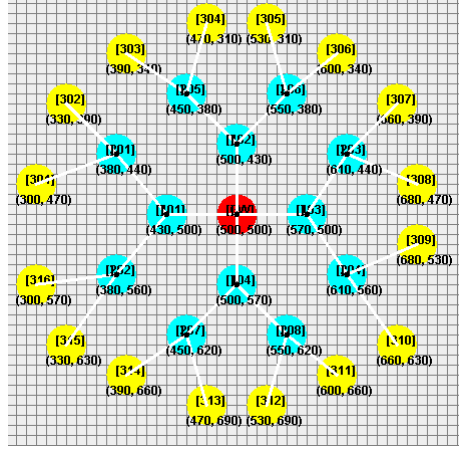


Fig. 4.12: Configuration 1: Network deployment with 28 sensor nodes and two children per node. (Two Children Per Node - C1 - 28SN)

3. Reduced load and congestion on the resource-constrained sensor nodes.
4. Increased delivery capability in case of network partition.

4.3.2 Wireless Sensor Network Configurations

The simulations are executed using a number of network configurations. Each network configuration has a different number of sensor nodes, some of which were configured as cluster heads and others as end devices. The cluster heads are responsible for forwarding data packets towards the destination from the end devices. The end devices are responsible for generating unique data packets, assumed to contain sensor readings. All the network configurations, however, only have a single fixed gateway, assumed to be connected to a remote command and control server, as different personal area networks (PAN) are executed typically on separate radio channel, resulting in independent performance. For each network configuration, the same variations of the number of mobile devices (0,2,5,10,15,20,50,80) were used for evaluation, resulting in eight combinations per network configuration.

The first network configuration (Figure 4.12) just had two children for every cluster

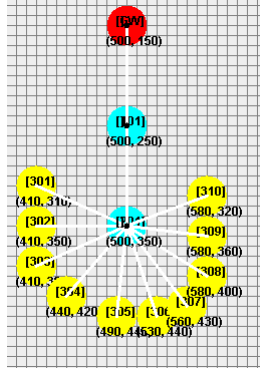


Fig. 4.13: Configuration 2: Single branch network deployment with 12 sensor nodes.
(Single Branch - C2 - 12SN)

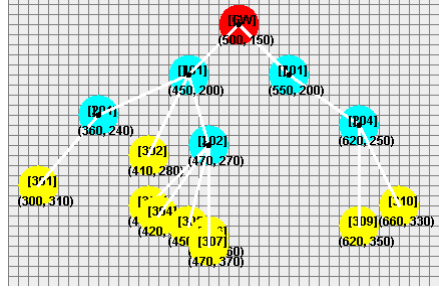


Fig. 4.14: Configuration 3: Unbalanced network deployment with 14 sensor nodes.
(Unbalanced Network - C3 - 14SN)

head. In total, this network configuration had 28 sensor nodes. This network configuration was used as a starting point for the evaluation.

The second network configuration (Figure 4.13) was designed with just a single branch with ten end devices generating sensor data every 15 seconds. This configuration was used to induce higher incoming traffic load on a single cluster head.

The third network configuration (Figure 4.14) was an unbalanced (asymmetric) network configuration, such that one branch of the network had more sensor nodes and end devices than the other main branch of the network. This enabled the study of the effect of the STERIOD protocol in asymmetric networks.

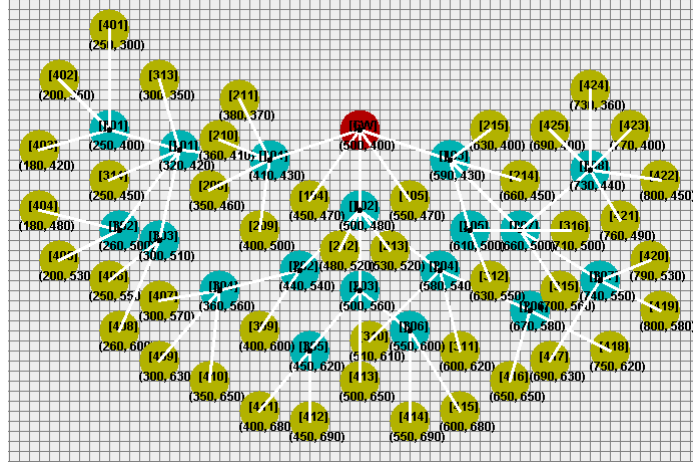
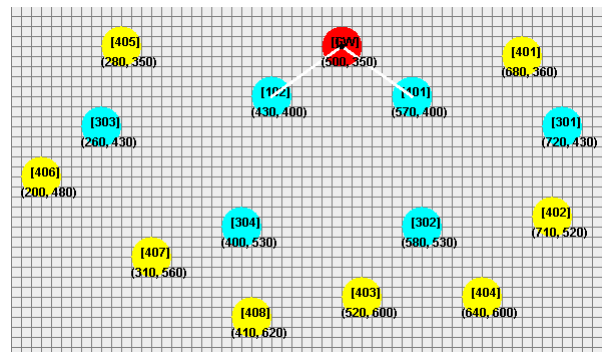
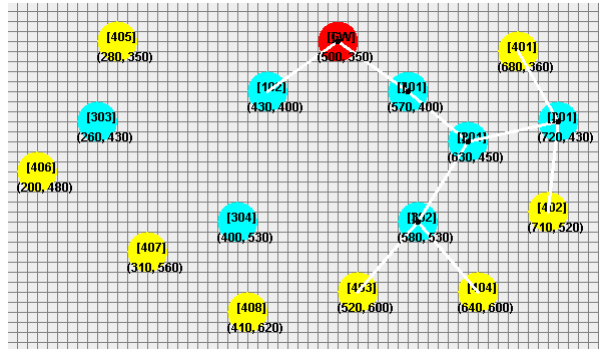


Fig. 4.15: Configuration 4: Planned hierarchical network deployment with 61 sensor nodes. (Hierarchical Designed Network - C4 - 61SN)

The fourth network configuration (Figure 4.15) was planned to simulate a more complicated hierarchical network topology. This network configuration had a maximum depth of four hops from the fixed gateway.

In order to evaluate the capabilities of the STERIOD protocol in terms of fault tolerance and bridging the gap in a partitioned network, two different network configurations were used. Configuration 5 (Figure 4.16) is half connected, i.e., one branch of the network was connected, whereas the other branch is disconnected because the connecting cluster head was assumed to have depleted its energy budget. Without the mobile devices, the delivery rate is expected to be exactly 50% for this network configuration. Configuration 6 (Figure 4.17) is completely disconnected, and thus its initial delivery rate is expected to be 0%. With the presence of mobile devices, the delivery rate is expected to rise to close to 100%, provided that enough mobile devices are present in the vicinity of the sensor network.

The last network configuration (Figure 4.18) is generated algorithmically in a circular pattern. This network configuration has a maximum depth of five hops. Some nodes in this network configuration are expected to experience higher volumes of incoming



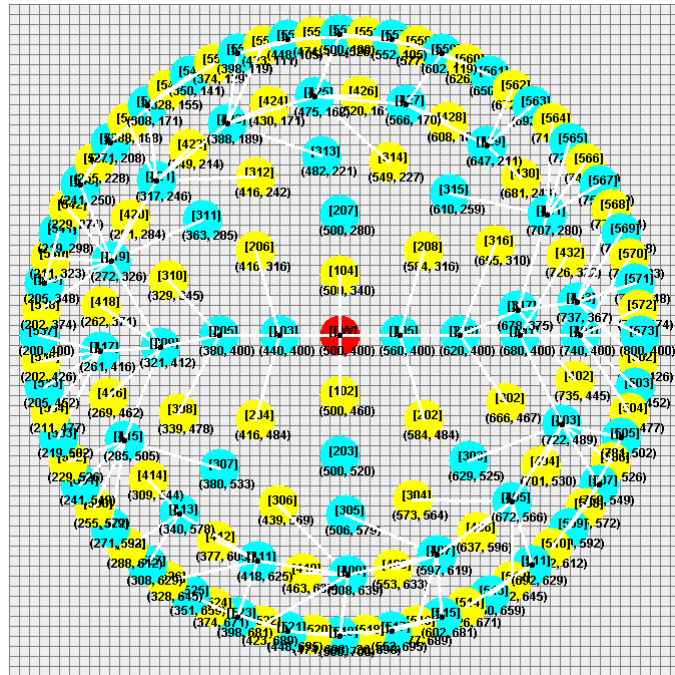


Fig. 4.18: Configuration 7: Semi automatic circular network deployment with 137 sensor nodes. (Semi Automatic Circular - C7 - 137SN)

data traffic than the others. Due to congestion and queue overflow, the delivery ratio is expected to be below an optimum value of 100% even without the presence of the mobile devices and the associated increased data loss risk.

4.3.3 Data Loss Due to Mobility And Its Remedy

When utilizing mobile devices, there exists a risk of packet loss. The packet loss depends on the trajectory that the mobile device takes, the speed at which the mobile device travels and the timing of the packet transfer.

Packet loss due to mobility can be reduced if the data transfer starts quickly as the mobile device arrives in range, i.e., the mobile device is discovered quickly, which improves the chances that the data transfer ends before the mobile device moves out of range. While this thesis aims at reducing the discovery interval, the sensor nodes transferring their data to the mobile device may not know when the mobile device will move out of range, because the sensor nodes do not know mobile devices' trajectories and speeds. For this reason, some data loss is still expected to occur.

In order to further reduce the data loss and keep the protocol light-weight, especially at sensor nodes, a retry mechanism is implemented at the sensor nodes. According to this mechanism if no acknowledgement is received, then the sensor node will retry sending the packet via its parent node, given that such a parent node exists.

In order to study the effect of this retry mechanism, the network deployments of Configuration C1 and C2 are used. For each network deployment, the sensor nodes were programmed to retry transmission if they do not receive an acknowledgement in one instance and in the other instance they do not retry.

As this experiment was designed to influence the design of the routing protocol, only four runs per network deployment were conducted to get a sense of the effect of the retry mechanism.

In this experiment, the following metrics were measured:

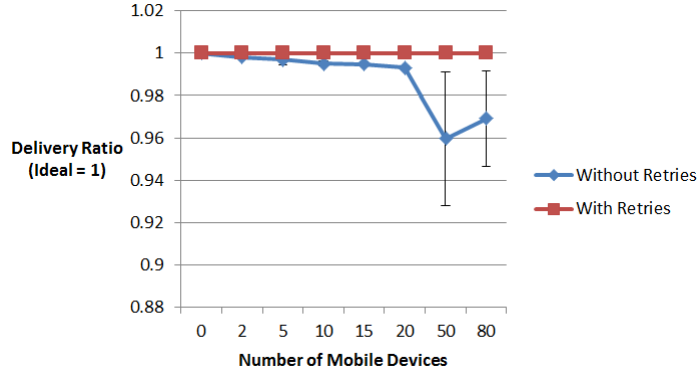


Fig. 4.19: Delivery ratio, with and without transmission retries, using network configuration C1.

1. Delivery Ratio, given by $\frac{\text{Number of Packets Delivered}}{\text{Number of Packets Generated}}$
2. Transmissions required by sensor nodes per generated packet, given by $\frac{\text{Packets Transmitted by Sensors}}{\text{Number of Packets Generated}}$.
3. Transmission overhead ratio per generated packet, given by $\frac{\text{Packets Transmitted by Sensors} + \text{Advertisements Sent by Sensors} - \text{Number of Packets Generated}}{\text{Packets Transmitted by Sensors} + \text{Advertisements Sent by Sensors}}$.

As the number of mobile devices increase, the mobility present in the network also increases. Due to this high mobility, the chance of packet drop also increases.

As expected, Figures 4.19 and 4.20, both show a general drop in the delivery ratio as the number of mobile devices increase. However, Figure 4.20 also shows that the delivery ratio is much lower when no mobile devices are present in the network, irrespective of the presence of retries. This observation also shows that the delivery ratio improves with the increase in the number of mobile devices. This phenomenon is observed because when no mobile devices are present in the vicinity of the sensor network, the only cluster head connected to all the end devices, which are generating sensor data, gets overwhelmed by the number of packets received. The incoming reception queue in this case gets overflowed in the absence of mobile devices. However, as the number of mobile devices increases,

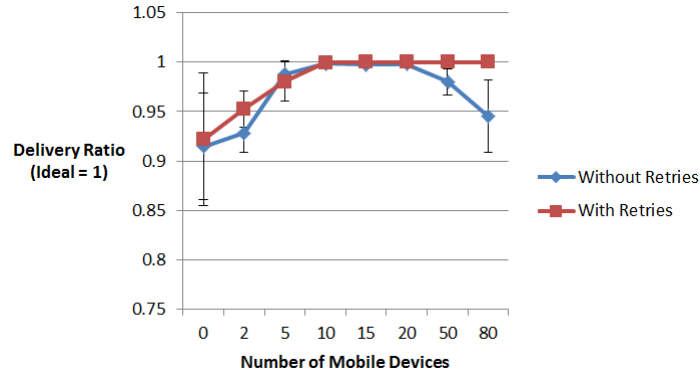


Fig. 4.20: Delivery ratio, with and without transmission retries, using network configuration C2.

the load on the cluster head decreases, and therefore more packets are delivered to the destination through the alternative path.

It can also be seen that when the retry mechanism is not implemented, the delivery ratio on average drops to almost 95%, as the number of mobile nodes increases. However, if the retry mechanism is implemented, only a very small fraction of the generated packets are lost when the number of mobile devices is varied from zero to 80. As the sensor nodes retry sending packets via their parent nodes when their transmission failed via mobile device, the delivery ratio is reduced mainly due to queue overflows at the parent nodes. These results show that there is a definite advantage in implementing the retry mechanism.

However, at the same time, there is a small disadvantage in terms of retransmissions. This disadvantage is evident from Figures 4.21 and 4.22, where a marginally higher number of transmissions required by sensors and a marginally higher transmission overhead ratio per generated packet are observed when the retry mechanism is implemented for the network configuration C2. This result may vary depending on the network topology and the trajectory of mobile devices. But the general conclusion remains the same, that the overhead on the sensor nodes will always be higher if a retry mechanism is

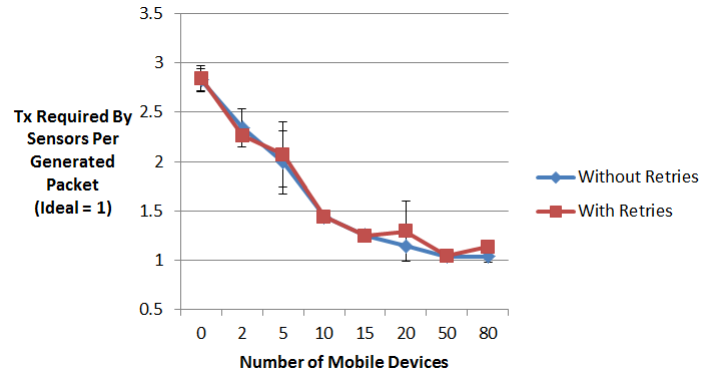


Fig. 4.21: Transmission required by sensors per generated packet, with and without transmission retries, using network configuration C2.

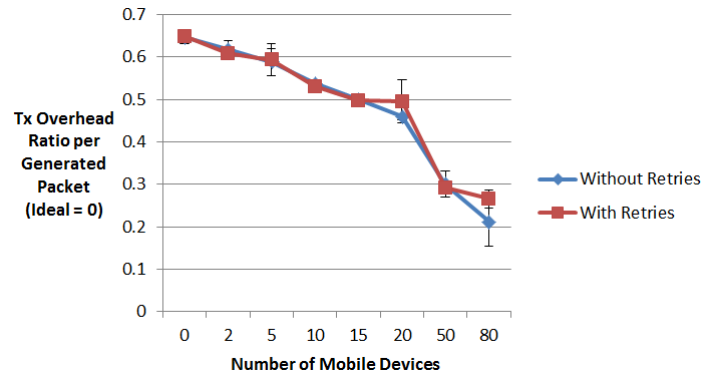


Fig. 4.22: Transmission overhead ratio per generated packet, with and without transmission retries, using network configuration C2.

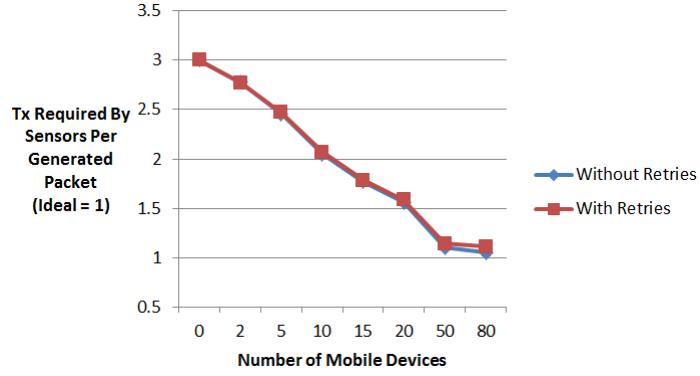


Fig. 4.23: Transmission required by sensors per generated packet, with and without transmission retries, using network configuration C1.

implemented on the sensor nodes.

If the network configuration is simple, e.g., network configuration C1, the increase in overhead is not observable for a smaller number of mobile devices. This phenomenon is evident from Figures 4.23 and 4.24.

As this work aims to reduce the communication load on the resource-constrained sensor nodes, it is essential to consider the impact of retries on the resource-constrained sensor nodes. It is evident that the presence of mobile devices can reduce the transmissions by the sensor nodes as the need to forward packets is reduced, however, the number of transmissions and the overhead observed at the sensors per generated packet, are increased when retries are implemented.

These results show a trade-off between increasing the delivery ratio and reducing the communication load on resource-constrained sensor nodes, as hypothesized in Chapter 1, Section 1.4. With a small increase in the communication load on sensor nodes, the increase in delivery ratio does seem to outweigh this disadvantage. As the number of mobile devices in the sensor network increases, the number of packets transmitted by the sensor nodes reduces significantly, while maintaining the delivery ratio, even in the presence of the retry mechanism. Thus, a clear advantage is observed in using mobile

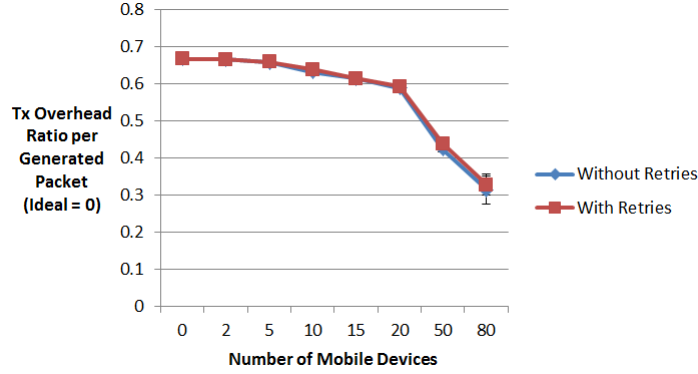


Fig. 4.24: Transmission overhead ratio per generated packet, with and without transmission retries, using network configuration C1.

devices to offload some communication load away from the sensor nodes and towards more capable mobile devices.

Further experiments will show that an increasing number of mobile devices also lowers the average number of hops that a packet traverses, hence reducing end-to-end latency (Section 4.3.5). In addition, as the communication load on sensor nodes is shifted towards more capable mobile devices, the energy consumption on the sensor nodes is also expected to reduce.

For these reasons, the retry mechanism was integrated in the STERIOD protocol.

4.3.4 Effect of the Selection Mechanism

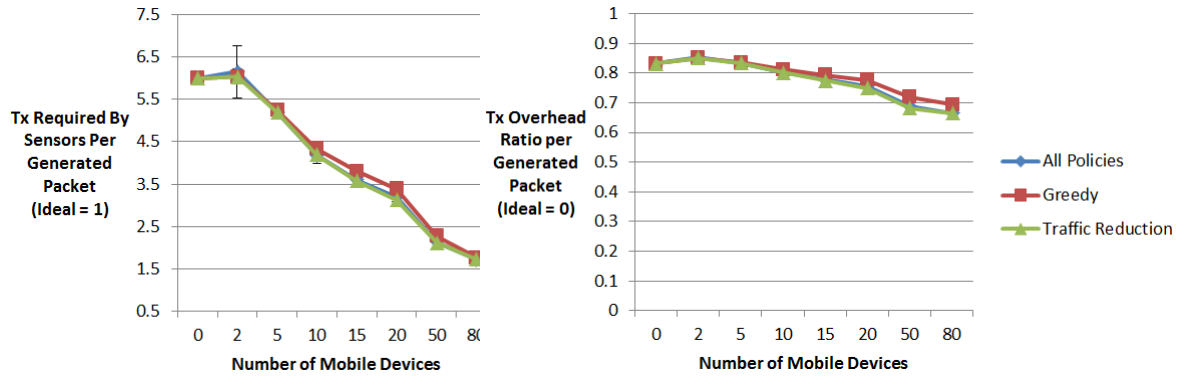
Chapter 3 (Section 3.4.5) described the mechanism for selecting a sensor node by the mobile device. The selection mechanism takes the decision according to the condition of the sensors in the vicinity of the mobile device in order to achieve one of the prioritized objectives. The action associated with each objective is triggered if the sensors in the vicinity of the mobile device meet the triggering condition for the respective objective. Triggering an action based on the condition of the sensor network in order to achieve a certain objective, is referred to as a policy. However, apart from the triggering strategy,

each policy is based on one of the two selection mechanisms (Chapter 3, Table 3.2), namely:

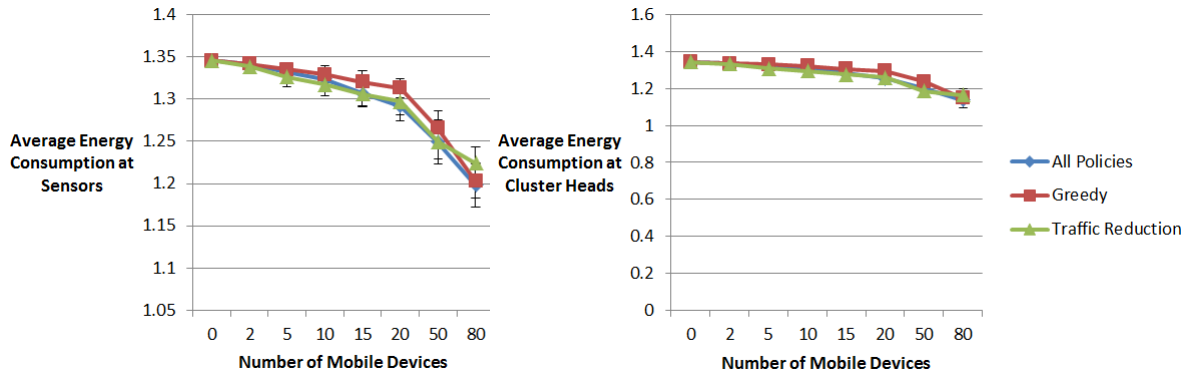
1. Credential improvement
2. Traffic reduction

In order to evaluate the effect of the selection mechanisms, a single selection mechanism was configured in the simulation at a time for each network configuration. In these cases, a pre-selection trigger and post-processing strategy were not used. Additionally, in another set of experiments, all policies along with the pre-selection trigger and post-processing strategy were configured in the simulation. These three sets of experiments allowed the evaluation of the individual selection mechanisms versus the policy based selection mechanism, which triggers individual selection mechanism based on network condition to achieve the prioritized objectives.

In the most congested network configuration with a high data traffic (C7), the traffic reduction selection mechanism shows better results when compared to the credential improvement selection mechanism. As evident in Figure 4.25, the traffic reduction selection mechanism required a smaller number of transmissions by the sensor nodes per generated packet (Figure 4.25a), a smaller overhead ratio per generated packet (Figure 4.25b), and a smaller energy consumption at the sensors and cluster heads alike (Figures 4.25c and 4.25d). Even the number of hops traversed on average was slightly lower (Figure 4.26a), when the traffic reduction selection mechanism was used. However, for the same network configuration, generally the delivery ratio was also lower (Figure 4.26b) when the traffic reduction selection mechanism was used. This shows that within the same network configuration, not all objectives were fulfilled by the traffic reduction selection mechanism. The credential improvement selection mechanism (marked as “Greedy” in the graphs) showed better delivery ratios, even if in other metrics, the results were not the best of the discussed mechanisms.



(a) Transmissions required by sensors per generated packet. (b) Transmission overhead ratio per generated packet.



(c) Energy consumption at sensors. (d) Energy consumption at cluster heads.

Fig. 4.25: Results using different selection mechanisms showing better performance with the traffic reduction selection mechanism, using the network configuration C7.

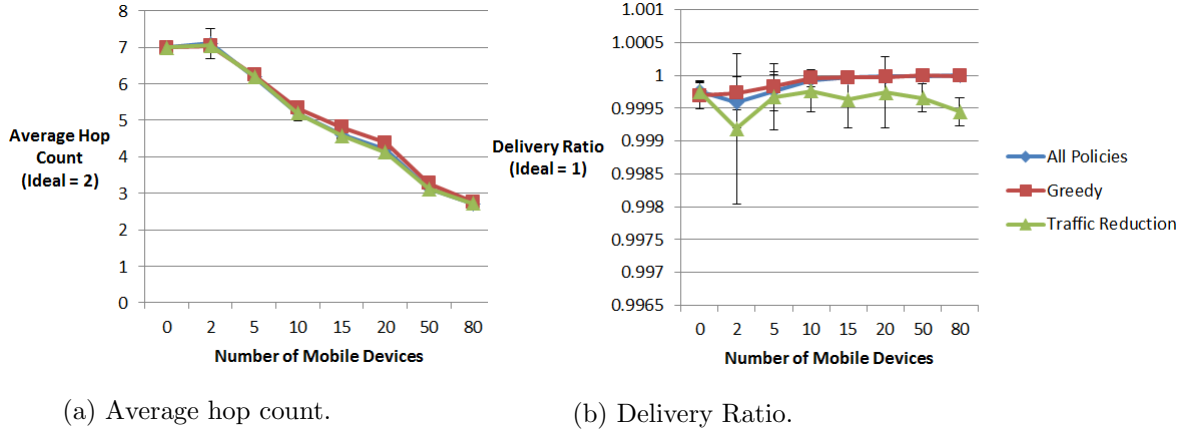


Fig. 4.26: Results using different selection mechanisms showing better average hop counts with the traffic reduction mechanism but better delivery ratios with the credential improvement selection mechanism, using the network configuration C7.

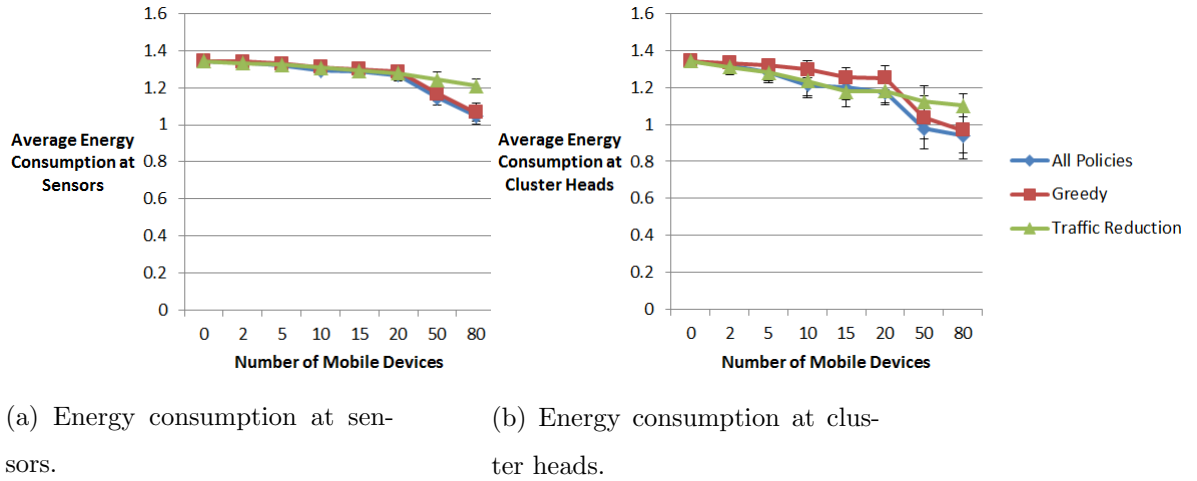
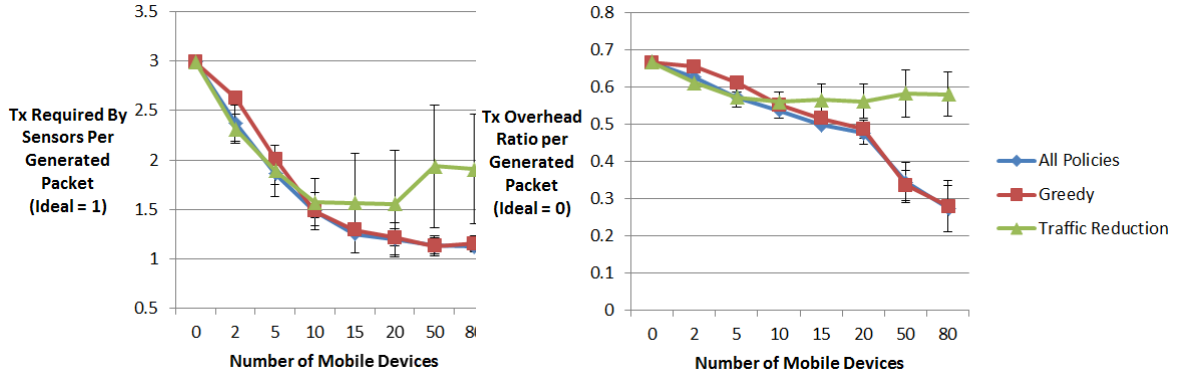


Fig. 4.27: Energy consumption using different selection mechanisms, using the network configuration C4.



(a) Transmissions required by sensors per generated packet. (b) Transmission overhead ratio per generated packet.

Fig. 4.28: Transmissions required by the sensor and transmission overhead ratio per generated packet using different selection mechanisms, using the network configuration C2.

Similarly using the network configuration C4, Figure 4.27 shows that better energy conservation at the cluster heads is achieved for lower number of mobile devices when the traffic reduction selection mechanism is used. Using the credential improvement selection mechanism for higher numbers of mobile devices resulted, however, in better energy conservation at the cluster heads. Not only that, but overall better energy conservation is achieved at the sensors with the credential improvement selection policy, when a high number of mobile devices are used, whereas for lower numbers of mobile devices, the energy conservation is similar in both cases.

The results from the network configuration C2, also imply that the performance of the two selection mechanisms can invert (sometimes drastically) within the same network configuration but with different number of mobile devices that are present in the vicinity. Figure 4.28 shows that a smaller number of transmissions are required at the sensor nodes per generated packet when the traffic reduction selection mechanism was used for lower numbers of mobile devices, however, for higher numbers of mobile

devices, the credential improvement selection mechanism performed much better. The behaviour observed from Figure 4.28b for the transmission overhead ratio per generated packet, also conveys a similar story.

These results show that a single selection mechanism is certainly not ideal to achieve all objectives, even within the same network configuration. Without knowing the network configuration a priori and without executing any performance tests before the deployment of the actual selection mechanisms, implementing a specific selection mechanism is unsuitable. From these results, the challenge therefore became apparent, i.e., to find a way to trigger the correct selection mechanism without assuming much knowledge about the network configuration. Thus, a triggering mechanism was devised, so that the performance could remain close to optimum in all network configurations, as discussed in Chapter 3 (Section 3.4.5.5).

When all policies are used by triggering a selection mechanism based on the network conditions, better results are observed throughout. The policy associated with prioritized objectives achievement, show smaller overheads and numbers of transmissions required by sensors per generated packet, smaller energy consumption, fewer traversed hops and higher delivery ratio, across all network configurations. Some of these results are evident from Figures 4.25, 4.26, 4.27 and 4.28. In these results, the plot associated with “All Policies” refers to the triggering mechanism based on the network state. In all cases, this plot coincides with the best result from any of the two individual selection mechanism.

4.3.5 Effect of Mobile Devices on the End-to-End Latency

In this research, the lowest priority is given to latency improvement, i.e., the mobile device tries to reduce latency (number of hops) when none of the conditions associated with the other objectives meet. Even with the lowest priority, the results obtained by running multiple simulation runs using a number of different network configurations, still show an interesting trend.

Figure 4.29 shows the average number of hops that packets traverse in order to reach

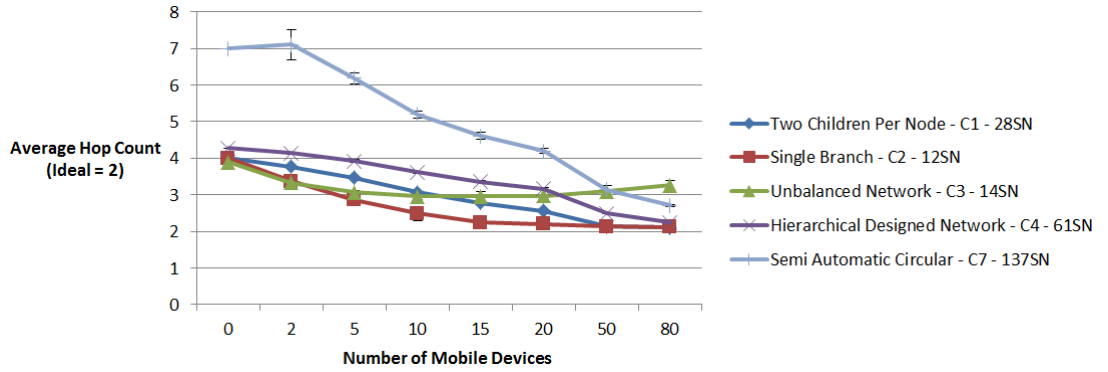


Fig. 4.29: Average number of hop counts traversed by generated packets, using connected network configurations C1, C2, C3, C4 and C7.

the final detination, i.e., the centralized command and control server. This graph shows the average number of hops traversed by the packets for different numbers of mobile devices present in the vicinity of the sensor network. The graph is generated using five different connected network configurations (C1, C2, C3, C4 and C7). Each time a packet is transmitted, the hop counter for the respective packet is incremented.

When no mobile devices are present in the vicinity of the sensor network, the packets traverse a fixed number of hops towards the destination. Of course, the number of hops depends on the network configuration, and thus a very high number of hops is shown when the network configuration C7 is used.

As the destination is assumed to be located at a distant server, i.e., behind an IP network, the ideal number of hops is expected to be two. This means that the shortest path between a sensor node that generated a packet and the final destination is two hops away, such that an intermediary device (such as Fixed Gateway) is required to bridge the WSN with the IP network.

Figure 4.29 shows a downward trend, irrespective of the network configuration, that reaches towards the ideal value of two hops as the number of mobile devices increases in the network. However, as the retransmissions increase due to high mobility, an increase

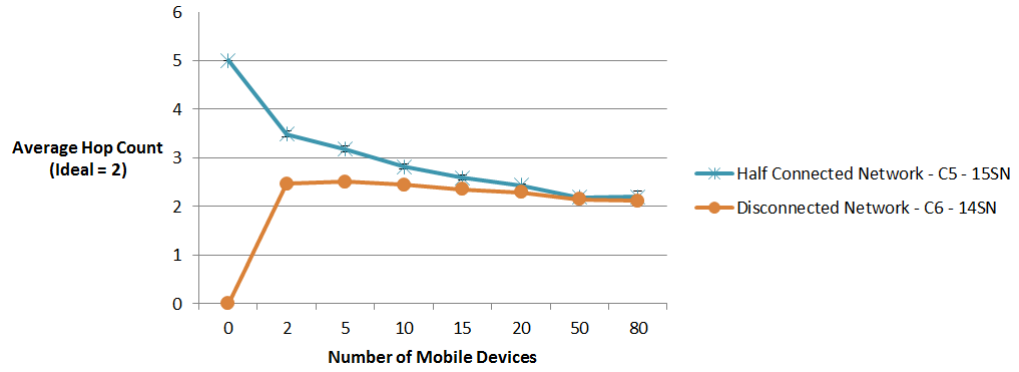


Fig. 4.30: Average number of hop counts traversed by generated packets, using network configurations C5 and C6.

in the number of hops can also occur, as evident for the network configuration C3. In general, the trend seems to stabilize near the ideal value as the ratio between the number of stationary sensor nodes and mobile devices approaches one.

Figure 4.30 shows the effect of increasing the number of mobile devices on the average hop count in network configurations that are experiencing partitions. For the network configuration C5, a non-zero number of hops traversed are shown because half of the network is connected to the final destination via a multi-hop route, when no mobile devices are present. However, for the network configuration C6, when no mobile devices are present, the average number of hop count is zero. This is due to the fact that none of the packets were delivered in this case, because all the end devices responsible for generating sensor readings are disconnected from the final destination.

Figure 4.30 shows that even for a partitioned network, as the mobile devices are increased in the vicinity of the sensor network, the average number of hops traversed by the generated packets converges towards the ideal value of two hops. In the completely disconnected network, the average number of hops can also be witnessed above two hops, because the mobile devices may connect to a cluster head as it may be the closest sensor node available to them. In turn, the cluster head connected to the mobile device

may indirectly provide a temporary route to its children via the mobile device. This phenomenon will be discussed further later on in this chapter.

4.3.6 Effect of Mobile Devices on the Communication Load

The objective to reduce congestion at sensor nodes experiencing a high volume of incoming traffic is triggered when the number of packets transmitted in the last minute by a sensor node is higher than the median of the number of packets transmitted by all other sensor nodes that are present in the vicinity of the mobile device (Chapter 3, Section 3.4.5).

In order to measure the communication load on the resource-constrained sensor nodes with respect to the overall communication activities in the network, the following metrics were measured:

1. The transmission ratio by sensors, given by $\frac{PacketsTransmittedbySensors}{TotalPacketsTransmitted}$, showing the proportion of the overall load handled by sensors.
2. The transmission ratio by cluster heads, given by $\frac{PacketsTransmittedbyClusterHeads}{PacketsTransmittedbySensors}$, showing the proportion of the overall load on all the resource-constrained sensors that is handled by cluster heads.
3. The number of transmissions required by the sensor nodes per generated packet, given by $\frac{PacketsTransmittedbySensors}{NumberOfPacketsGenerated}$, showing the load on sensors for each generated packet.
4. The transmission overhead ratio per generated packet, given by $\frac{PacketsTransmittedbySensors+AdvertisementsSentbySensors-NumberOfPacketsGenerated}{PacketsTransmittedbySensors+AdvertisementsSentbySensors}$, showing the overhead incurred by sensors apart from the generated packets.

For this experiment, only the fully-connected network configurations (C1, C2, C3, C4 and C7) were used, as fault tolerance is evaluated separately.

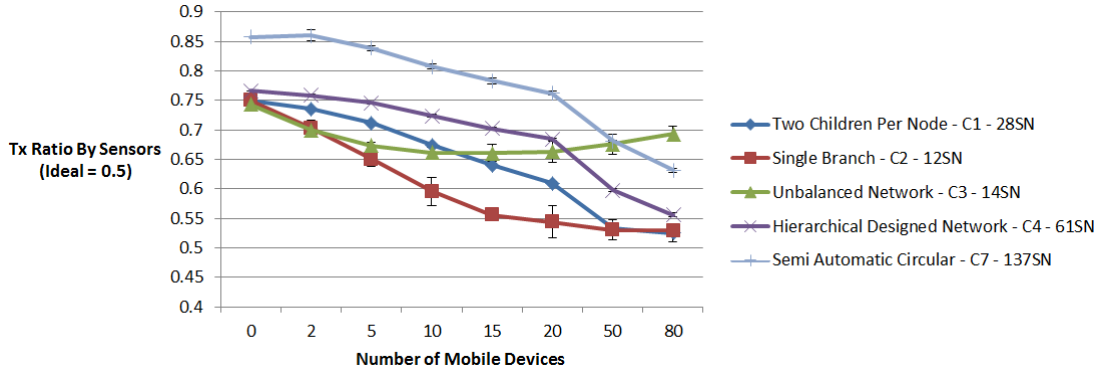


Fig. 4.31: Transmission ratio by sensors, using the network configurations C1, C2, C3, C4 and C7.

Ideally, if an end device that is responsible for generating sensor data is connected to the final destination (command and control server), only two transmissions should be required to reach the destination, as discussed in the previous section. Thus, the goal is to reduce the transmission ratio by sensors to a value of 0.5 reflecting two overall transmissions required to reach the destination. For this reason, Figure 4.31 shows an ideal value of 0.5. As can be seen for almost all network configurations considered in this experiment, when the number of mobile devices are increased in the network, the transmission ratio by sensor nodes shows a downward trend towards the ideal value. As the number of mobile devices increases and the network stabilizes, this transmission ratio also approximately stabilizes near the ideal value. An improvement of around 20% can be observed in ideal cases, whereas in the absence of mobile devices, the transmission ratio by sensors reached up to 85% in the worst case, suggesting that most of the communication was carried out by resource-constrained sensor nodes.

In this research, the cluster heads are considered to act only as intermediate devices that provide a path towards the destination, i.e., they are assumed not to generate their own unique data packets. In an ideal case, all the communication load on the cluster heads will be offloaded to more capable mobile devices, resulting in the transmission

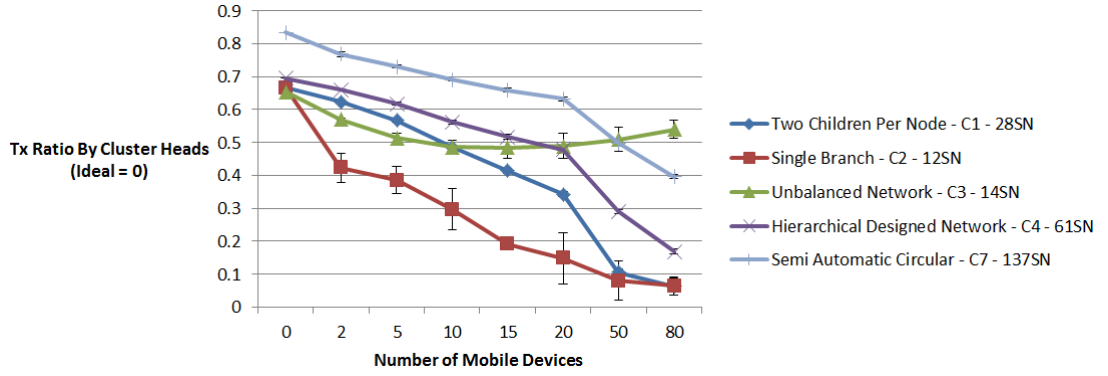


Fig. 4.32: Transmission ratio by cluster heads, using network configurations C1, C2, C3, C4 and C7.

ratio by cluster heads to reach an optimum value of zero. Figure 4.32 shows that the transmission ratio by cluster heads reduces dramatically in almost all the network configurations considered in this experiment. The biggest improvement is seen in the network configuration C2, as the transmission ratio by cluster heads is reduced much faster than for all other network configurations and reaches close to a 10% value, from a value of 66% when no mobile devices were present. The semi-automatic network deployment, i.e., configuration C7 also shows a huge improvement, as when 80 mobile devices are used, the transmission ratio by cluster heads drops to almost 40%.

As discussed above, in an ideal scenario, the end devices will directly transmit their sensor data to a fixed gateway to be transported to the final destination. In this ideal case, only a single transmission is required by the sensor nodes to deliver data. For this reason, Figure 4.33 shows an ideal value of one for transmissions required by sensor nodes per generated packet, assuming that all data packets are delivered. As more mobile devices appear in the vicinity of the sensor network, irrespective of the network configuration, this metric also converges towards the ideal value. Figure 4.33 shows the highest observable gain for the network configuration C7, where on average as much as six transmissions were required for each generated packet, but with just 5 mobile devices,

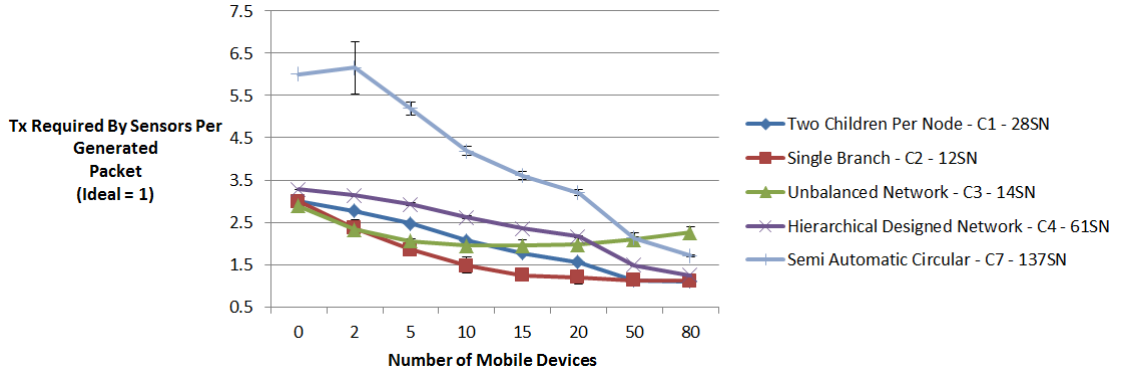


Fig. 4.33: Transmissions required by sensors per generated packet, using network configurations C1, C2, C3, C4 and 7.

this reduced to almost 5 transmissions by sensors for each generated packet, eventually reducing to a value of 1.7 on average, with 80 mobile devices.

Even though the ideal value of the transmission overhead ratio by sensor per generated packet is set at zero, it is impossible to achieve in a multi-hop network configuration. The ideal value of zero reflects that no transmission overhead should be observed, such that no advertisement packets are transmitted, and the only transmissions performed by the sensor nodes are the transmissions carried out to transmit the generated packets by the end device(s). If no mobile devices is available in a multi-hop network, the overhead will occur due to re-transmissions via cluster heads. If mobile devices are available, and all the packets are forwarded via the mobile device from the end device, then the overhead will occur due to advertisement packets being generated as a consequence of the recursive credential update mechanism discussed in Chapter 3, Section 3.4.2. Nevertheless, Figure 4.34 shows a general reduction in the transmission overhead by sensors with increasing number of mobile devices. As expected, a higher overhead reduction is observed for some network configuration than for others.

Apart from the general trends, the network configuration C3 stands out, as for this configuration each of the above mentioned graphs shows an eventual increase in the

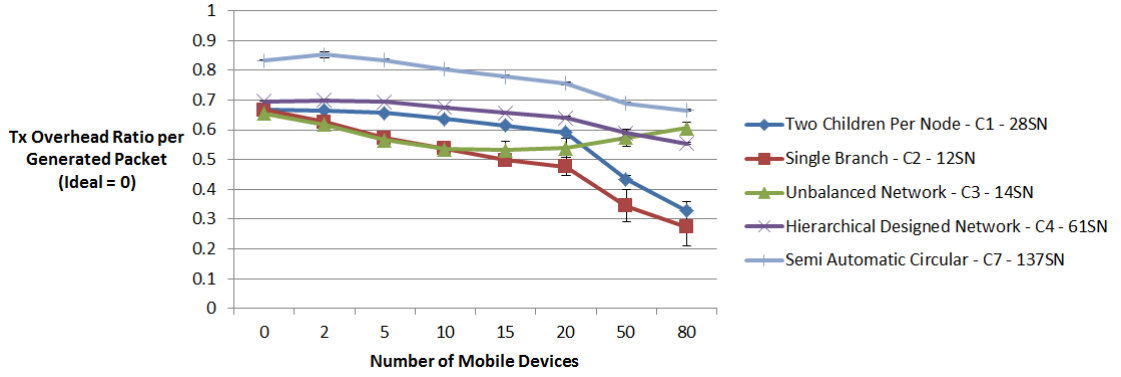


Fig. 4.34: Transmissions overhead ratio by sensors per generated packet, using network configurations C1, C2, C3, C4 and C7.

transmission ratio/overhead at the sensors. This phenomenon may have been caused by the re-transmissions due to the mobility of the mobile devices. Even though the performance in this network configuration is not ideal, the communication load on the resource-constrained sensor nodes is reduced, when compared to the same network configuration with no mobile devices.

4.3.7 Effect of Mobile Devices on Fault Tolerance

In this research, the objective of achieving fault tolerance in case of network partitions was given the highest priority, such that if a network partition is detected, then the mobile devices will try to provide paths to the destination to those devices that are experiencing disconnection from the rest of the network. A sensor node that is not connected to the network will, by design, have close to zero credentials to reach the destination. Thus, this objective is triggered whenever a node is detected that has a credential value of less than 30. Having a lower credential threshold of 30 for activating the fault tolerance objective achievement, not only provides a path to sensor nodes that are disconnected from the network, but also provides alternative paths to sensor nodes that are far from the destination.

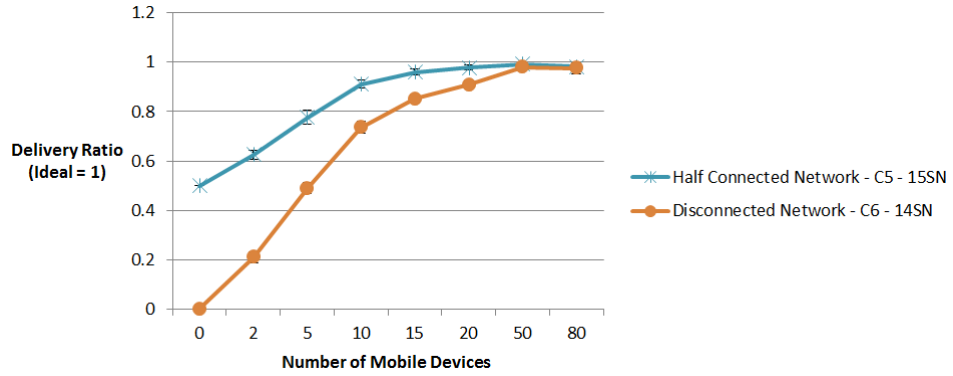


Fig. 4.35: Delivery ratio, using the network configurations C5 and C6.

The most important characteristic to measure for this experiment is the delivery ratio in partitioned networks. For this experiment, the network configurations C5 and C6 (half-connected and disconnected networks) are used.

As expected, Figure 4.35 shows that when no mobile devices are present in the network, the delivery ratio is exactly what is expected, i.e., the half connected network configuration is only able to deliver 50% of the packets, whereas the completely disconnected network configuration is unable to deliver any packets at all. With the introduction of mobile devices in the network, the delivery ratio steadily increases, ultimately reaching close to 100% when 50 mobile devices are available in the network vicinity for both network configurations. With even higher number of mobile devices (80), a slight drop in the delivery ratio can be observed. This drop can be explained by failed attempts to route data via mobile devices in the presence of high mobility.

As the mobile devices arrive in the vicinity of the network, some of them may advertise themselves to the cluster heads due to their proximity. Once the cluster heads connect to the mobile devices, their credentials improve, allowing end devices to select these cluster heads that are temporarily connected to the destination via mobile devices to forward their sensor readings.

This phenomenon is evident in Figure 4.36, especially for the completely disconnected

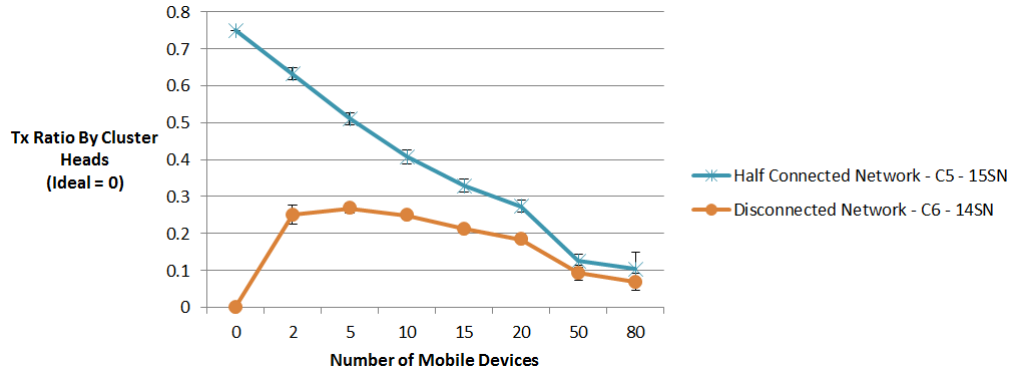


Fig. 4.36: Transmission ratio by cluster heads, using the network configurations 5 and 6.

network configuration C6, such that the transmission ratio by cluster heads, given by $\frac{PacketsTransmittedbyClusterHeads}{PacketsTransmittedbySensors}$, increases as the mobile devices are introduced in the network. As the number of mobile devices increases in this network configuration, the transmission ratio at the cluster heads reduces again. This pattern shows that not only the protocol allows packets to be delivered via cluster heads by providing alternative paths to the destination through mobile devices, the protocol also reduces the load on the cluster heads as the mobile devices increase in the network.

For the network configuration C5, even though only half the network was connected to the destination, about 80% of the packets were delivered via cluster heads. As the number of mobile devices present in the network increases, the load on cluster heads is reduced significantly to almost 10% (Figure 4.36), in addition to increasing the delivery ratio (Figure 4.35) at the same time.

Figure 4.37 shows the number of transmissions required by sensor nodes per generated packet, given by $\frac{PacketsTransmittedbySensors}{NumberofPacketsGenerated}$, for the partitioned network configurations. When no mobile devices are present, no transmissions by sensor nodes are observed in the fully partitioned network configuration (C6), because the end devices did not transmit the generated packets due to the unavailability of paths towards the destination. As the

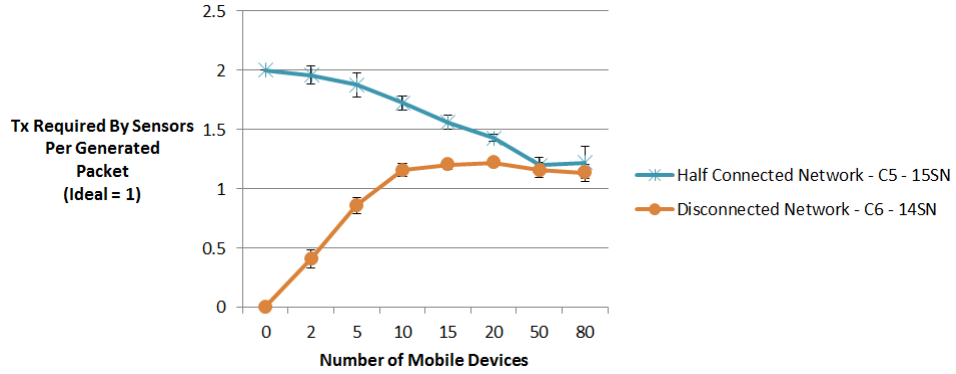


Fig. 4.37: Transmissions required by sensors per generated packet, using the network configurations 5 and 6.

number of mobile devices present in the network increases, this transmission ratio by sensors per generated packet, first increases and later stabilizes close to the ideal value of one. For the network configuration C5, the transmission ratio by sensors reduces steadily until reaching the value of 1.2, that is again very close to the ideal value.

Even though the transmission ratio by sensors should be as small as possible, the value below the ideal value of one is misleading. When less than 10 mobile devices are present in the network, for the fully partitioned network configuration, this ratio is below the ideal value of one. This phenomenon only means that not all the generated packets are being delivered to the destination, as is evident from Figure 4.35, for the same network configuration and the number of mobile devices. In fact, the only time a transmission ratio by sensors is expected to be below the ideal value of one is when the delivery ratio is either less than or equal to 50%, and all the packets are delivered directly either via mobile devices or fixed gateway. This means that not all the end devices have a path to the destination, and therefore, they do not actually transmit their generated data packets.

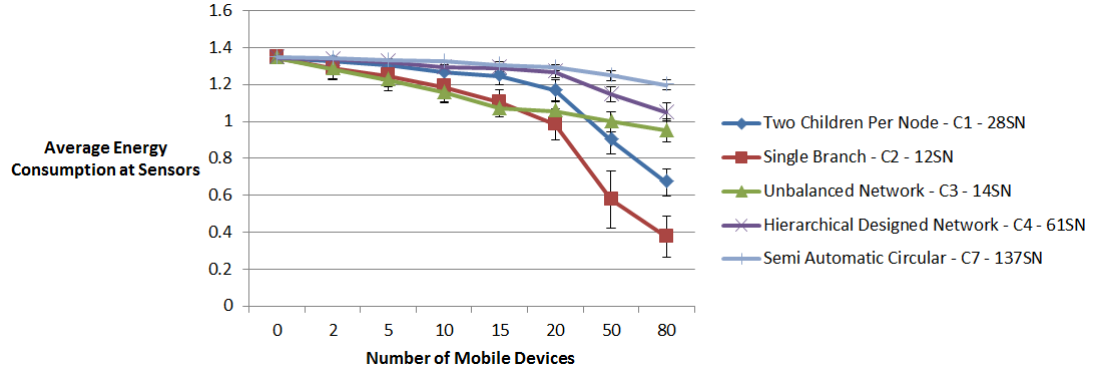


Fig. 4.38: Energy consumption at sensor nodes in milliamperes, using network configurations C1, C2, C3, C4 and 7.

4.3.8 Effect of Mobile Devices on Energy Consumption

In order to evaluate the energy consumption on the sensor nodes, each node was initially provided with an energy budget of 1 milli ampere hour and the simulations were executed for 50 minutes, as discussed in Section 4.2.5.

As seen in Figure 4.38, the average energy consumption on each sensor node with a constant beacon interval, data rate and packet size, is almost 1.4 milli-amperes during the course of the simulation, with no mobile devices, irrespective of the network configuration. This clearly shows that on average each node has surpassed its energy budget (the node death, however, was not simulated).

As the mobile devices are introduced in the network, the mobile devices trigger their energy conservation objective for each cluster head whose normalized energy reserves have depleted beyond a 65% lower-threshold value. In addition to the energy conservation objective, the congestion reduction and latency reduction prioritized objectives are also triggered depending on the network conditions.

Figure 4.38 clearly shows that the energy budget is significantly conserved during the course of the simulation as a higher number of mobile devices are introduced in the network. However, it is also noteworthy that different levels of energy conservation occur

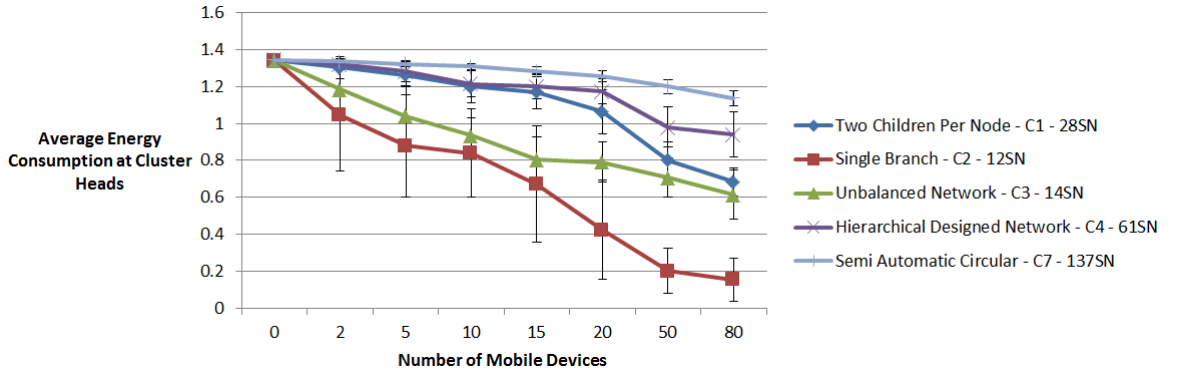


Fig. 4.39: Energy consumption at cluster heads in milliamperes, using network configurations C1, C2, C3, C4 and C7.

in different network deployment scenarios. The best case scenario reveals energy savings of almost 70%. However, in the worst case, a reduction in energy consumption of only about 10% is achieved.

Figure 4.39 shows similar trends of energy consumption at the cluster heads alone. However, each plot shows higher improvement on cluster heads on average as compared to all the sensors. In the best case scenario, energy savings of almost 90% are observed, whereas in the worst case, energy savings of around 16% are observed on the cluster heads.

These observations clearly show the achievement of a significant reduction in energy consumption in each of the network deployments, using the STERIOD protocol. In addition, as the protocol reduces the incoming load on nodes that are showing high transmission, such as the cluster heads, the results in this section show more improvement on such critical nodes that are required to maintain long term connectivity of the sensor nodes with the destination. Higher energy conservation is achievable if more mobile devices appear in the vicinity of the sensor nodes.

4.3.9 Energy Consumption of Mobile Devices

Energy consumption on the smart mobile devices was not of direct interest in this thesis, as the main objective of this research was to reduce communication load and energy consumption on the sensor nodes, in addition to providing fault tolerance and reducing end-to-end latency. This section briefly discusses the energy consumption on the mobile devices, nevertheless.

When vehicles are considered as mobile devices, the impact of energy consumption is irrelevant, as the vehicles charge their battery when driven, thus no energy depletion will be observed in normal gasoline based vehicles. In battery operated vehicles, such as smart cars, the energy consumption due to STERIOD and ILAD protocol will remain insignificant compared to their primary functions, i.e., driving the car itself. Energy depletion is seen as a concern for individually owned smart mobile devices, such as smart phones, tablets and laptop computers. Within this category as well, the most interesting aspect of energy depletion is seen on the hand-held devices, i.e., phones and tablets, as they are used when in motion, sometimes the software work in the background on such devices without the manual intervention of the individual user.

Balasubramanian et al. (2009) discuss the effect of energy consumption with respect to different modules on mobile devices. The authors show that different radio modules on a hand-held smart phone have different energy consumption for the same amount of data transfer. They show that 3G consumes most energy when compared to GSM and WiFi data transfers. GSM consumes less energy than WiFi when also considering WiFi association up to 100 kilobytes of data transfer, but the energy consumption on GSM exponentially increases after the 100 KB mark. However, the energy consumption of WiFi for data transfer alone is comparatively significantly more efficient than GSM or 3G modules.

This shows a common observation that the lifetime of the hand-held smart mobile device after a full recharge depends on its usage by the individual. In fact, if GPS is

switched on, significant energy consumption can be observed. These observations are also validated by the author of this thesis as discussed briefly under Appendix B.4.

When the STERIOD and ILAD protocols actually get integrated into real smart phones, the impact on energy consumption on the smart phones is expected to be limited. The reason being that the ILAD discovery mechanism will only enable radio channel scanning when it expects sensor nodes to be in the vicinity of the smart mobile device. Whereas, the location will be fetched by the mobile device only when allowed by the owner of the device. Thus, when the individual user switches on either GPS, 3G or WiFi, the location will also become available to the STERIOD and ILAD protocols. The energy consumption by the actual radio module that will communicate with sensor nodes is also expected to remain low as the STERIOD routing protocol and ILAD mechanism primarily focus on using low-power MAC layer protocol such as IEEE802.15.4 (2006) to communicate with the sensor nodes.

4.3.10 Effect of the Speed of Mobile Devices

As seen in the previous sections, a penalty is associated with introducing mobile devices in the network. Indeed due to the mobility of the mobile devices, packets routed via mobile devices can be frequently lost. In the previous experiments, the random way point model was used with zero pause time and the maximum speed was configured at 10 units/second. In order to study the impact of speed on the performance of the protocol, separate experiments were conducted with different maximum speeds (4, 8, 12, 16 units/second) in the random way point model, with different numbers of mobile devices.

Figure 4.40 uses the network configuration C7 to show that the delivery ratio drops as the configured maximum speed is increased. This phenomenon was especially evident when few mobile devices were present in the network. In the presence of retry mechanism, discussed in Section 4.3.3, the drop in the delivery ratio was insignificant, delivering almost 100% of the packets to the destination.

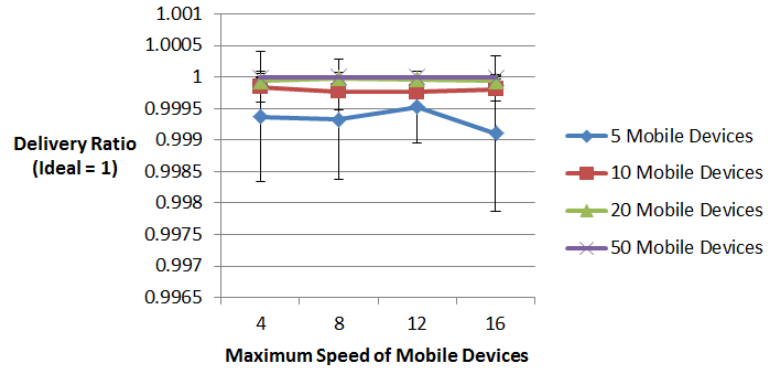


Fig. 4.40: Delivery ratio for different maximum speed and mobile devices, using the network configuration C7.

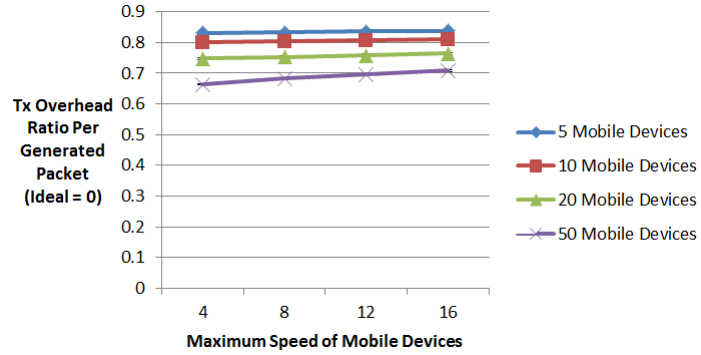


Fig. 4.41: Transmission overhead ratio per generated packet for different maximum speed and mobile devices, using the network configuration C7.

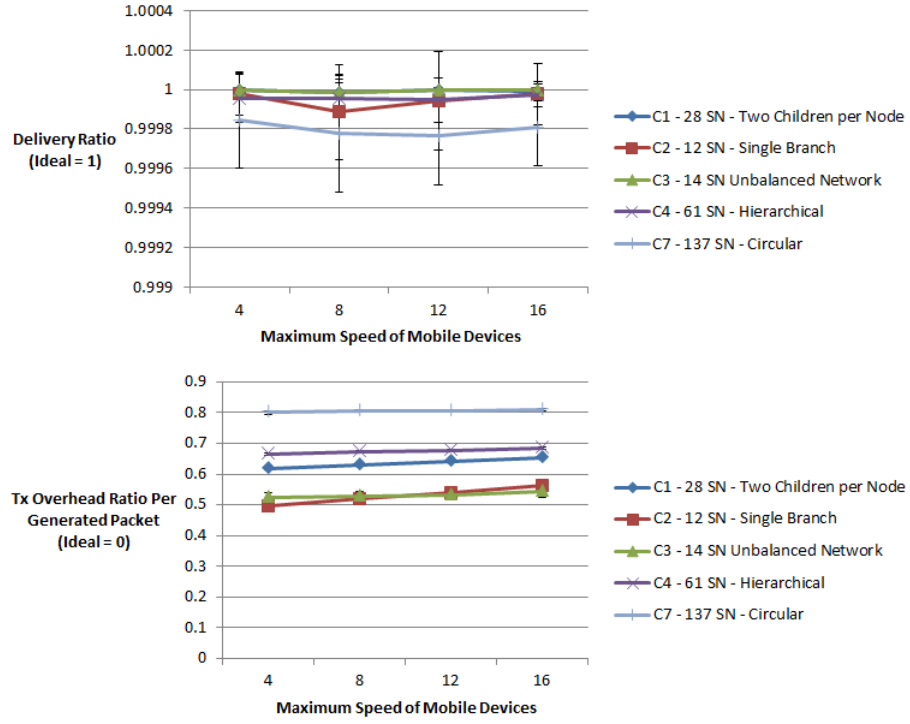


Fig. 4.42: Delivery ratio and transmission overhead ratio per generated packet for different maximum speed with 10 mobile devices, using the network configurations C1, C2, C3, C4 and C7.

However, Figure 4.41 reveals that the retry mechanism indeed incurred higher transmission overheads when the maximum speed of the mobile devices is increased. While maintaining the delivery ratio, the transmission overhead, given by

$$\frac{\text{PacketsTransmittedbySensors} + \text{AdvertisementsSentbySensors} - \text{NumberofPacketsGenerated}}{\text{PacketsTransmittedbySensors} + \text{AdvertisementsSentbySensors}},$$

increases due to a higher number of transmissions by the sensor nodes and a higher number of advertisements generated due to frequent topology changes. The increase in transmission overhead is evident with a higher number of mobile devices available in the network.

Figures 4.42 and 4.43, show the same phenomenon for the network configurations C1, C2, C3, C4 and C7.

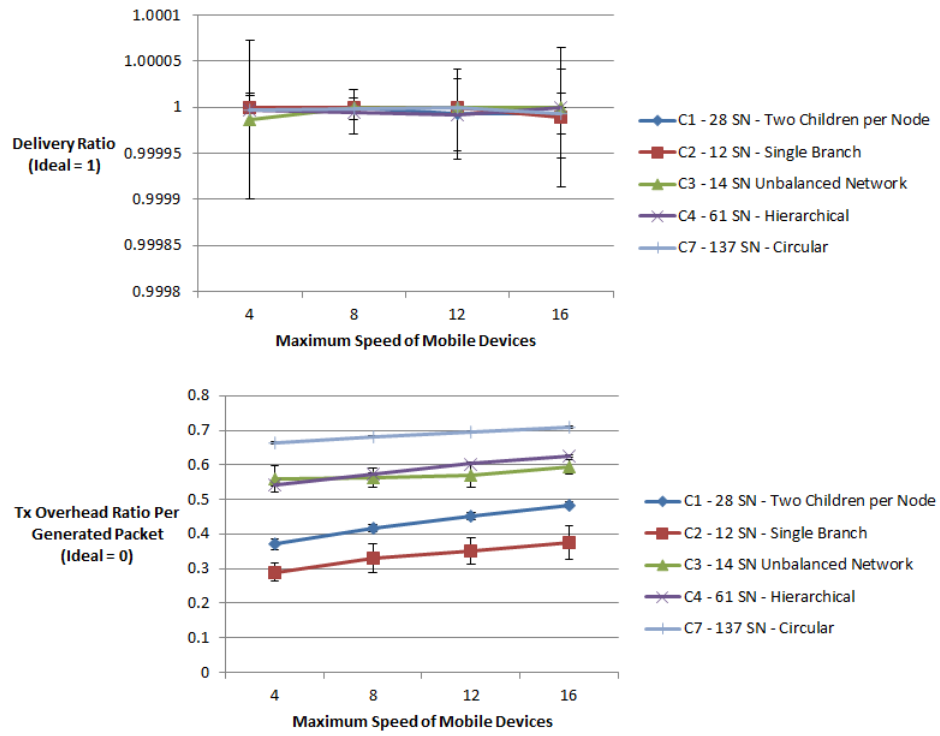


Fig. 4.43: Delivery ratio and transmission overhead ratio per generated packet for different maximum speed with 50 mobile devices, using the network configurations C1, C2, C3, C4 and C7.

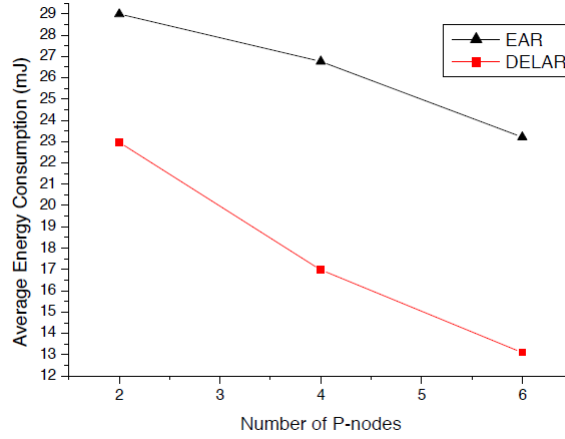


Fig. 4.44: Energy consumption with increasing number of P-nodes (Liu et al., 2011).

The delivery ratio is confirmed to reduce across all the considered connected network configurations, as the maximum speed of the mobile devices is increased. Also, the transmission overhead ratio also increases due to retries, across all network configurations.

When comparing results from Figures 4.42 and 4.43, another interesting phenomenon is observed, where the average transmission overhead tends to be comparatively lower and the average delivery ratio is higher when more mobile devices are present for the same speed and network configuration.

Similar observations are also reported by existing work that utilizes mobile devices (Liu et al., 2011; Du et al., 2006), such that a reduction in delivery ratio and an increase in the overhead is observed with the increasing speed of the mobile devices.

4.3.11 Comparative Study

The closest and the most recent work related to this research was discussed by Liu et al. (2011), which introduced the “Device-Energy-Load Aware Relaying (DELAR)” framework for heterogeneous mobile ad hoc networks, in which the traffic was redirected towards more capable nodes (P-nodes) in order to reduce the traffic load and energy consumption on less capable nodes in the network. Liu et al. (2011) assumed that all

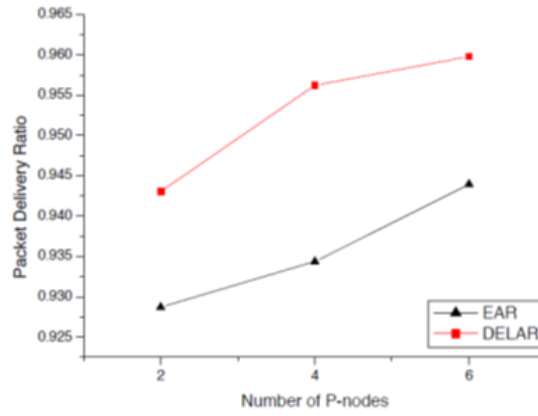


Fig. 4.45: Delivery ratio with increasing number of P-nodes (Liu et al., 2011).

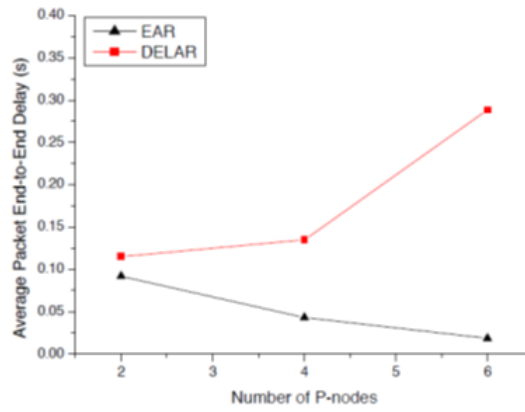


Fig. 4.46: Average end-to-end delay with increasing number of P-nodes (Liu et al., 2011).

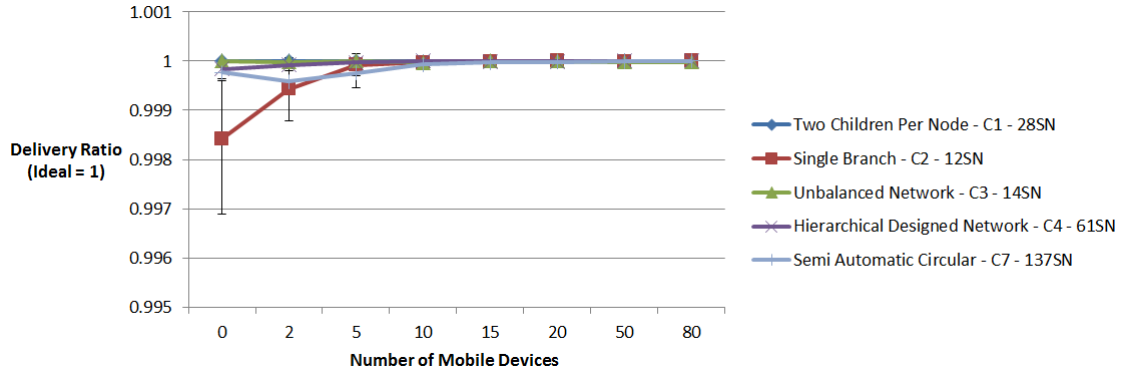


Fig. 4.47: The delivery ratio with increasing number of mobile nodes, using the network configurations C1, C2, C3, C4 and C7.

the nodes in the network are mobile with zero pause time in their simulations. The network nodes were randomly deployed and followed a random way point model to simulate mobility.

A direct comparative study of the DELAR protocol (Liu et al., 2011) against the STERIOD protocol, introduced in this research, that would allow the comparison within the same environmental setting is not present, primarily due to the unavailability of the implementation of the DELAR protocol. This section, however, compares the results available from DELAR (Liu et al., 2011) and those obtained for the STERIOD protocol.

Even though Liu et al. (2011) did not specifically target stationary WSNs with mobile devices, they showed that introducing more capable devices in the network reduced the overall energy consumption (Figure 4.44). It is expected that the energy consumption at the resource-constrained sensor nodes could be lower when using STERIOD than when using DELAR, because the less capable nodes in the DELAR protocol can only deactivate their radio in the allocated time period when more capable nodes communicate with each other, whereas with the STERIOD protocol, the less capable sensor node activate their radio only during the active portion of the superframe typically resulting in small listening duty cycle. In addition, the DELAR protocol is based on the proactive routing

protocol DSDV (Perkins and Bhagwat, 1994). Therefore, the drawbacks associated with traditional routing protocols (Chapter 2, Section 2.2.1) also apply, i.e., higher route repair activities when highly mobile devices are considered in the network.

Liu et al. (2011) compared their results against the energy aware routing (EAR) that is also based on DSDV (Perkins and Bhagwat, 1994). In addition, they also showed that the delivery ratio also increased (Figure 4.45) as the number of more capable nodes in the network increased, because the more capable nodes in their network had longer transmission ranges than the normal nodes (using an asymmetric MAC layer). These results are very similar to the results shown in this research in Figures 4.38 and 4.39, such that the STERIOD protocol also achieved significant energy savings with the introduction of more capable mobile devices. The STERIOD protocol also shows an increase in the delivery ratio with the increasing number of mobile devices, especially in partitioned (Figure 4.35) and congested network deployments (Figure 4.47).

However, as the DELAR protocol (Liu et al., 2011) allocates special slots to individual more capable nodes (P-nodes) in the superframe, the average end-to-end latency increases as the number of P-nodes increases in the network (Figure 4.46). Since no such special slots are allocated in this research, the increase in the number of more capable mobile devices has no such adverse effect. On the contrary, the end-to-end delay reduces as the number of more capable mobile devices increases in this research (Section 4.3.5). This reduction in end-to-end delay (represented by the reduction in average hop count) is achieved due to the fact that the more capable mobile devices offer shorter paths to the destination.

As the maximum speed of nodes is increased, the DELAR protocol reports higher end-to-end delays (Figure 4.48) and lower delivery ratio (Figure 4.49). Similar effects are also observed in this research, such that the delivery ratio reduces and the transmission overhead per generated packet increases with the increasing speed of more capable mobile devices (Figures 4.40, 4.41, 4.42, 4.43). However, due to the retry mechanism implemented in this research, the delivery ratio does not reduce too much, as packets

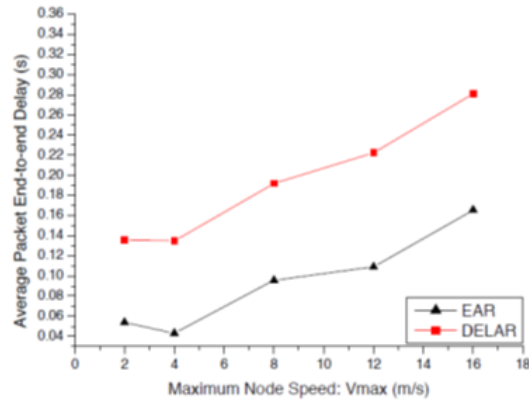


Fig. 4.48: Average end-to-end delay with increasing maximum speed of P-nodes (Liu et al., 2011).

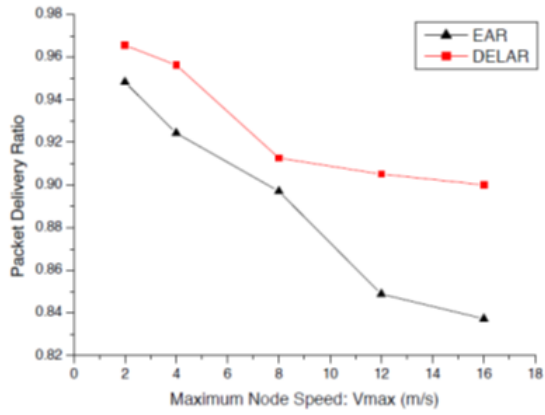


Fig. 4.49: Delivery ratio with increasing maximum speed of P-nodes (Liu et al., 2011).

are resent via fixed paths if the transmission via mobile devices fail (Section 4.3.3).

The similarities in the results discussed in this chapter and the results reported by Liu et al. (2011) corroborate the validity of the findings presented, whereas, the performance of the STERIOD protocol does not degrade with increasing number of mobile devices.

4.3.12 Discussion

The results shown in this chapter can successfully conclude that the protocol designed in this research performs well under a variety of circumstances, and indeed reduces the load on the stationary WSN. The STERIOD protocol does indeed reduce **(i)** the end-to-end delay by reducing average hop count, **(ii)** the traffic load on the less capable devices by reducing the incoming traffic load on the devices experiencing a high communication load, **(iii)** the energy consumption by outsourcing network communication via mobile devices and by utilizing the “Inverse Location-Aware Discovery”, and **(iv)** provides fault tolerance to the sensor nodes experiencing disconnection from the network.

The STERIOD protocol not only reduces the communication load on the cluster heads, but also on the end devices that may become intermediate devices as a result of the presence of mobile devices. Especially in the case of network partitions, as the mobile devices advertise themselves to the cluster heads, the protocol allows other sensor nodes in the vicinity of the cluster head to also take advantage of the mobile device indirectly. Similarly, the results show a reduction in the energy consumption across all the resource-constrained sensor nodes.

The results also show that having multiple policies responsible for achieving individual objectives based on the current network conditions performs much better across different network deployment scenarios than implementing a single mechanism to achieve the different objectives. When multiple policies are implemented, all the objectives are achievable simultaneously.

4.4 Summary

This chapter began by dividing the evaluation of this research into two parts. The first part evaluated the proposed “Inverse Location-Aware Discovery” mechanism against the conventional discovery mechanism discussed in Section 3.3.2 of Chapter 3. Later, this chapter evaluated the “Smart Traffic Energy and Resource aware Offloading using Inverse Discovery (STEROID)” routing protocol in detail and discussed different properties of the routing protocol.

4.4.1 Discovery Mechanism

The discovery mechanisms were evaluated using a very basic experimental environment to study the discovery problem in isolation, without using any routing protocol. The evaluation was carried out using a single sensor node and a single mobile device in a test-bed equipped with a sniffer. The mobility of the mobile device was emulated from a configuration file, so that the results can be reproduced, precisely. Initially, the trajectory of the mobile device followed only a single path, and then multiple paths were introduced through which the mobile device arrived in the vicinity of the sensor node. This method allowed the results to be generalized.

The evaluation showed that the distribution of the discovery time became smooth as more paths were introduced in the mobile device’s trajectory. In addition, for the discovery mechanisms discussed in Sections 3.3.2 and 3.3.3 of Chapter 3, similar distributions of the discovery time were witnessed. However, the discovery interval of the proposed ILAD mechanism, discussed in Section 3.3.4 of Chapter 3, shows that much lower bounds on the discovery time can be achieved if a location-aware discovery mechanism is used.

The discovery mechanism was also evaluated in terms of energy consumption at the sensor nodes. The results showed that, given the correct duty cycle is configured using the beacon and superframe orders, a lower energy consumption can be achieved when the mobile device discovers the sensor nodes, without penalizing the discovery time.

The results from the evaluation of the discovery mechanism are scalable as multiple sensor nodes will result in the same or similar discovery behaviour, as the evaluation is performed independently of the routing protocol. The results show that the proposed “Inverse Location Aware Discovery” mechanism is not only energy efficient but simultaneously discovers the sensor nodes faster than the conventional discovery mechanisms. The only drawback in this type of discovery is that it cannot be used in target tracking scenarios as the mobile devices do not transmit periodic beacons and only a single sensor node can directly communicate with a mobile device at a time, as the mobile device advertises to only a single sensor node. However, the benefits offered by this discovery mechanism in terms of energy conservation and fast reactivity at the sensor nodes, generally trumps the drawbacks.

4.4.2 Routing Protocol

This chapter evaluated the proposed “STEROID” routing protocol in a custom simulation environment in order to reduce the effort and complexity involved in executing multiple scenarios with different parameters and different network deployments, to study specific aspects of the protocol independently. The simulation environment implemented packet loss due to the mobility of the mobile devices and overflow of the reception and transmission queues at the resource-constrained sensor nodes, while also emulating mobility. The simulation environment also simulated energy consumption due to the transmission and reception tasks performed at the sensor nodes. The energy consumption was implemented in a way to emulate the energy consumption on the CC2420 (Chipcon, 2006) radio module. This chapter also discussed the limitations of the simulator developed as part of this research. Different network deployment scenarios were also discussed in this chapter, ranging from very simple to complex deployments, including deployments that simulated congested and partitioned networks.

This chapter discussed the data loss due to the mobility of the mobile devices and compared the performance of the routing protocol in the presence and absence of the

retry mechanism. The retry mechanism allowed the sensor nodes to resend the packets that are not acknowledged by the mobile device, via their parent node, if such a node exists. This chapter showed that such a mechanism reduced the data loss significantly at the cost of slightly increasing the communication load on resource-constrained sensor nodes, and was thus implemented in the protocol.

This chapter also argued that none of the individual selection mechanism available, obtained better results for different network configurations and different numbers of mobile devices. In certain situations, the credential improvement selection mechanism worked better, whereas, in other situations, the traffic reduction selection mechanism achieved better results. Better results were obtained in all network configurations and deployment scenarios, however, when all policies that triggered the right selection mechanism to achieve a given objective based on the network condition were implemented.

This chapter also empirically showed that the proposed routing protocol reduced the number of hops traversed with an increasing number of mobile devices, both in fully connected and disconnected network configurations. The proposed routing protocol significantly reduced the communication load on the sensor nodes experiencing high incoming traffic, in terms of transmission ratio by sensor nodes, transmissions required by sensor nodes per packet and transmission overhead that also included advertisement packets. In this chapter, the effect of utilizing mobile devices was also analysed in case of network partitions, using semi-disconnected and fully disconnected network deployments. The results showed that the delivery ratio significantly improved when mobile devices were utilized using the proposed routing protocol while simultaneously reducing the communication load on the sensor nodes. Furthermore, the results show that in a partitioned network, the communication load on the sensor nodes only increased when the sensor nodes were required to deliver data to the destination, from the nodes that did not have direct access to a mobile device. This chapter showed that apart from the energy savings from the proposed discovery mechanism alone, the routing protocol further reduced energy consumption on the resource-constrained sensor nodes, by redi-

recting traffic away from the sensor nodes that are low on energy budget. In addition, this chapter also discussed energy consumption on mobile devices in presence of different radio modules.

Next, this chapter analysed the effect of the maximum allowable speed of the mobile devices on the delivery ratio of the network, with varying numbers of mobile devices. The results showed a reduction in the successful packet delivery to the destination with increasing speeds of mobile devices. Due to the retry mechanism, only a small number of packets were lost, however, the transmission overhead per generated packet was increased. The results showed that the increase in the overhead was higher when higher numbers of mobile devices were present in the network. However, at lower speeds and for higher numbers of mobile devices, the transmission overhead was comparatively smaller.

A direct comparison with the DELAR (Liu et al., 2011) protocol in the same network setting was not performed, due to the unavailability of the protocol's implementation. However, this chapter showed that Liu et al. (2011) also reported similar results, in terms of the trends of energy consumption and packet delivery ratio, with the introduction of mobile devices. To some extent, the similarity in the trend of the results validate the simulation environment used in this research. Where Liu et al. (2011) reported an increase in end-to-end latency with the increasing number of mobile devices, this chapter showed a decrease in the number of hops (representing a reduction in latency) using the proposed routing protocol when mobile devices were introduced in the network. It was argued that the energy consumption at resource-constrained sensor nodes is expected to be lower with the STERIOD protocol when compared to the DELAR protocol, because the less capable nodes in the DELAR protocol can only deactivate their radio in the allocated time period when more capable nodes communicate with each other. In contrast, this research leverages the periodic inactive periods in the superframe from the IEEE802.15.4 (2006) MAC protocol, allowing the resource-constrained sensor nodes to deactivate their radios in order to conserve energy.

Chapter 5

Conclusions and Future Work

This thesis addressed the problem of routing sensor data packets in wireless sensor networks (WSNs) in the context of cities, in order to reliably offload communication tasks away from resource-constrained sensor nodes by opportunistically utilizing more capable mobile devices present in the vicinity of the sensor network.

This work argues that in addition to energy and load aware metrics, such decisions must be taken in a way that is energy efficient and reduces not only the communication but also the processing load on the resource-constrained sensor nodes, while keeping the implementation simple as well to fit within the available resources. This is achieved by using the ILAD discovery mechanism. Additionally, this thesis demonstrates how multiple objectives of fault tolerance, energy conservation, congestion reduction and latency improvement can be systematically achieved in a prioritized fashion while limiting the risk of data loss due to mobility and user's control of the third-party mobile devices.

This chapter summarizes the thesis and outlines the achievements of this work in the next section. Section 5.2 performs a short critical analysis of the work presented in this thesis. The contributions made to the state-of-the-art are discussed in Section 5.3. Finally, Section 5.4 concludes this chapter and the thesis by setting the future direction of this research.

5.1 Achievements

WSNs are composed of battery-powered, resource-constrained autonomous devices. These devices (sensor nodes) are typically deployed to collect environmental data such as temperature, humidity, light, radiation levels, carbon emissions etc. Very often the sensor nodes are deployed in hard-to-access locations such that their maintenance is non-trivial and costly.

In order to continuously receive environmental information from these sensor nodes and increase their operational life time, the sensor network itself must be designed in a way that is energy efficient such that it reduces the chances of draining the energy reserves of the battery-operated sensor nodes. Since most of the energy is consumed during the communication activity, generally the protocols are built to reduce communication tasks on the sensor nodes. However, the transmission range is usually limited due to the resource-constrained nature of the sensor nodes. For this reason, multi-hop routes are built from the source node towards the destination. In such cases, the sensor nodes become responsible for not only transmitting their own data but also for forwarding data on behalf of other sensor nodes.

While energy-aware (routing) protocols allow the sensor nodes to take energy efficient decisions while delivering data to the destination, the intermediary sensor nodes cannot be avoided. In urban environments, the problem of multi-hop routing via the sensor nodes can be mitigated by using ubiquitously-available mobile devices. However, successfully discovering the mobile devices and then being able to quickly use those devices in an energy-efficient manner is non-trivial due to the mobility and user's control of such devices.

This thesis showed that existing discovery mechanisms are typically resource intensive especially on the sensor nodes. For this reason, this thesis introduced the "Inverse Location-Aware" discovery mechanism where the cooperative mobile devices scan the radio channel to discover sensor nodes when they are expected in range.

In addition, this thesis also showed that the existing routing protocols are incapable of efficiently taking advantage of mobile devices to redirect traffic away from the resource-constrained sensor nodes. Because of the heavy reliance on route discovery and repair procedures and the associated energy cost, the protocols that tend to redirect traffic towards more capable devices are not suitable in the context of WSNs.

Thus, a new routing protocol was designed in this thesis that enables the mobile devices to efficiently offer their services to the resource-constrained sensor nodes. In this novel protocol, the mobile devices intelligently select appropriate sensor nodes to achieve fault tolerance, energy conservation, congestion reduction and latency improvement in a prioritized fashion.

The mobile device discovery mechanism and the routing protocol introduced in this research were designed in a way to keep expensive tasks, in terms of energy and memory consumption and processing load, on more capable mobile devices. The implementation was inspired by software engineering principles to dynamically inject correct selection strategies depending on the network conditions. In addition, as much of the implementation as possible is carried out such that it is portable and is not restricted to certain types of devices. The implementation itself is extensible in the sense that it allows the protocol to be easily modified and extended in the future while conforming to well-known and well-established object oriented design patterns. Also, the implementation is based on the IEEE-802.15.4 standard to allow sensor nodes from multiple vendors to be used while keeping the protocol agnostic to the hardware platform. In addition, to fit within the very constrained resources at WSN nodes, the implementation on the sensor nodes was designed with most of the complex tasks performed by the more capable devices.

The evaluation presented in this thesis showed that the proposed ILAD discovery mechanism is not only energy efficient at the sensor nodes but also allows the mobile devices to discover the sensor nodes quickly as it starts the discovery process only when it expects sensor nodes in its vicinity. Using seven different sensor network configurations, the evaluation of the proposed device-aware STERIOD routing protocol showed that it

improves the data delivery ratio especially in congested networks. The routing protocol was also shown to be capable of significantly reducing the energy consumption at the sensor nodes, the traffic congestion at the critical nodes, the overall routing overhead and the end-to-end latency by intelligently redirecting packets towards more capable mobile devices when possible. In addition, in the case of partitioned sensor networks, the routing protocol was shown to be fault tolerant such that if large number of mobile devices are present, the data delivery ratio almost reaches 100%.

5.2 Critical Analysis

In this thesis, a hierarchical network topology was assumed, as they have been shown to scale well. Nevertheless, if the network is deployed in a mesh topology, the data communication can still be offloaded to more capable mobile devices using the STERIOD protocol, as long as the sensor nodes beacon periodically, allowing them to be discovered by the mobile devices. Similarly, the experiments performed in this thesis used homogeneous sensor nodes, however, heterogeneous sensor nodes can also be used if they conform to the IEEE-802.15.4 standard. Indeed the main motivation behind using this standard was its usability across sensor nodes manufactured by different vendors.

Typically, the sensor networks deliver their data to a centralized location. However, a small number of multiple destinations can be accommodated in the STERIOD routing protocol, with individual credential values maintained for each destination.

In this research, the data queries are assumed to be delivered via the fixed gateway. While it is not within the scope of this work, mobile devices can also be used to deliver data queries from the command and control server to the sensor network. In this case, the command and control server can issue data queries to the registered mobile devices operating in the vicinity of the sensor network from where data is to be collected. The mobile devices can then forward these data queries when they arrive in the vicinity of the targeted sensor network.

The mobile devices that cooperate with the WSN, can also consume the sensor data for their own use, as the sensor data is forwarded via such mobile devices.

This thesis assumes that the location of the sensor nodes is recorded at the deployment time and shared with mobile devices. If this was not possible, the sensor nodes can register their location with the centralized command and control server when a path is established. Once the location of the sensor nodes is recorded at the command and control server, these locations can be consumed by the mobile devices to make smart decisions depending on the network conditions, as discussed in this thesis.

This thesis assumes that the transmission range is known a priori. However, the known transmission range may not reflect the actual reachability of two sensor nodes due to multipath fading and channel interference. The known transmission range is used only by the mobile device to determine which two sensor nodes are in range of each other after the mobile device has received beacons (containing location information) from both sensor nodes during its scan cycle. This information is used to estimate parent-child relationships and make smart choices at the mobile device in order to achieve the prioritized objectives. While the effect of multipath fading and channel interference will be studied in the future work of this research, the impact of such radio characteristics is considered to be small on the decision making process carried out by the mobile device, because apart from the location information the mobile device also considers the credentials of the sensor nodes while identifying sensor nodes that are dependent on each other. The credentials of the sensor nodes are expected to reflect the reality of the connection between the two sensor nodes, as the credentials degrade by a pre-configured factor if a sensor node depends on another sensor node to deliver its data to the destination.

5.3 Contributions

The research described in this thesis contributes to the state-of-the-art in the area of routing and mobile device discovery within the context of urban-scale WSNs.

The core contributions of this thesis are as follows:

1. An extensive analysis of the state-of-the-art mobile device discovery protocols within the context of WSNs, and closely related routing protocols that utilize mobile devices in different capacities encompassing multiple domains of mobile ad hoc networks, delay tolerant networks and wireless sensor networks was performed. The analysis of the related work shows that the existing protocols cannot be used in resource-constrained WSNs to either discover mobile devices quickly and energy-efficiently, or to offload the communication and processing load to more capable mobile devices to achieve energy conservation, fault tolerance, congestion reduction and end-to-end latency reduction simultaneously while reducing the risk of data loss.
2. A new location-aware device discovery protocol (the Inverse Location-Aware Discovery - ILAD) moves the discovery procedure to the more capable mobile devices and allows sensor nodes and mobile devices to discover each other efficiently. It achieves faster discovery by exploiting location awareness at the mobile devices and improved energy consumption at the sensor nodes when compared to the baseline discovery protocol. An implementation of the discovery mechanism is provided on sensor nodes in TinyOS (Levis and Gay, 2009) on the TelosB (Crossbow, 2005) platform and on mobile devices in Java. The evaluation of the ILAD mechanism in a test-bed environment shows that it allows the mobile devices to quickly discover the sensor nodes while remaining energy efficient at the resource-constrained sensor nodes.
3. A new routing protocol (the Smart Traffic Energy and Resource aware Offloading

using Inverse Discovery - STERIOD) allows the more capable mobile devices to take over data forwarding activity from sensor nodes by exploiting ILAD. This routing protocol reduces end-to-end latency by avoiding multi-hop routes via sensor nodes, reduces congestion and energy consumption at intermediary nodes by reducing packet arrivals at the intermediary nodes and provides fault tolerance in the case of network partitions via mobile devices. An implementation of the routing protocol is provided in a custom Java based simulator that is portable on multiple operating systems. The evaluation of the STERIOD protocol in a simulated environment shows that the protocol indeed increases the packet delivery ratio by intelligently utilizing the mobile devices, especially in congested and disconnected sensor networks. The effect of the routing protocol is studied on factors such as transmission overhead, delivery ratio, transmissions required to deliver packets and transmissions by intermediate sensor nodes under different network deployment scenarios, showing that the STERIOD protocol meets all the goals and requirements set out in Chapter 1 under Section 1.5.

In conclusion, this work has shown how the next hop can be found in a WSN to move data forwarding and processing load to more capable mobile devices while simultaneously conserving energy, reducing congestion, providing fault tolerance, improving end-to-end latency and reducing the risk of data loss due to the mobility of the mobile devices. Additionally, this work has also shown how mobile devices can be quickly discovered while keeping the discovery mechanism energy-efficient at the resource-constrained sensor nodes.

5.4 Future Work

Mobile devices were used in this research to offload data communication from resource-constrained sensor nodes. Multiple mobile devices can select and offer their services to the same sensor node, but only one of the mobile devices will be used by the sen-

sor node. This inefficiency can be mitigated by incorporating coordination among the mobile devices, such that the mobile devices may optimize sensor node selection in a way that increases their usability throughout the sensor network. The future work of this research can focus on the coordination of the mobile devices without increasing the communication overhead too much.

Currently, reducing the energy consumption at the mobile devices was not the focus of this research because the mobile devices can typically be recharged easily. However, the future work of this research can also focus on evaluating and reducing the energy usage at mobile devices, and in particular usage of the localization mechanism based on the GPS module that typically incur large energy consumption at the mobile devices. Instead predicting the arrival of the mobile devices in the vicinity of the sensor nodes based on the temporal patterns of the mobile device can be explored as an alternative approach.

Additionally, the mobile devices were assumed to be cooperative in this research. In the future, incentivization of the mobile device to allow sensor nodes to offload their communication load can also be explored. Providing real world incentives such as coupons, free talk time, free data usage etc, to the mobile device owners is envisaged to be a promising direction to motivate mobile device owners, as also discussed in Chapter 1, Section 1.3.

A simple TCP/IP communication model was used for establishing communication between the command and control server and the fixed gateways and mobile devices. Future work can also focus on the server-side communication between these high-end devices to incorporate the service-oriented communication paradigm using SOAP and/or RESTful web services for communicating sensor data from the fixed gateway and mobile devices to the command and control server. Additionally, the publish-subscribe communication paradigm can be used to deliver geographic sensor data queries from the command and control server to the fixed gateway and mobile devices. Using such modern communication techniques will allow future implementations of this work to

communicate in more restricted networks where the communication ports other than HTTP (port 80) may be blocked, delivering a more robust solution.

This research used a risk metric per device type to reduce the risk of data loss. The future work of this research can include link quality estimation as part of the fixed network formation metric, potentially building on the existing work presented in Chapter 2 under Section 2.2.3.

The future work can also include store-carry-forward based approaches, inspired by the delay tolerant networking protocols, so that the sensor nodes may store their packets until a mobile device advertises itself, if the application requirements allow it. This will allow the STERIOD protocol to improve the energy conservation on the sensor nodes further as even less packets will be forwarded via the neighbouring sensor nodes.

The evaluation of this research can be extended in the future to include real-world evaluation of the STERIOD protocol based on real-world mobility and communication environment with the presence of radio interference from external sources.

Finally, efficient approaches to consolidate feedback from multiple sensor nodes will be explored, such that past experiences from a subset of sensor nodes may be shared with the rest of the sensor network. However, the associated communication overhead will need to be carefully assessed.

5.5 Summary

This chapter summarised the motivations for, and the most significant achievements of the work described in this thesis. In particular, it outlined how this work contributes to the state-of-the-art of the discovery of mobile devices and offloading communication load to mobile devices in urban-scale WSNs. In addition, a critical analysis of the work presented in this thesis was also performed in this chapter, and some suggestions for future work were presented.

Appendices

Appendix A

Design Techniques

The following sections discuss a few additional mechanisms incorporated in the design of the STERIOD routing protocol for improved and robust performance of the protocol.

A.1 Avoiding beacon collisions

When multiple devices are broadcasting periodic beacons, it is imperative to have a collision avoidance mechanism in place in order to avoid collisions among beacons of neighbouring devices.

Even though the CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) mechanism is available in many MAC layer protocols, collisions among beacons from neighbouring nodes may still occur (Wan et al., 2003) due to detection delay. For example, if two devices decide to transmit packet at the same time, and there are no other transmissions on the radio channel at that particular moment, then both devices may transmit their packets, sensing free channel. Unfortunately, in this case, collision is bound to happen resulting from the transmissions of the two devices.

For this reason, the sensor nodes divide their beacon transmissions in the time domain such that beacons from two devices do not collide with, and corrupt with each other.

In normal conditions when the network is formed, utilizing a cluster-tree network

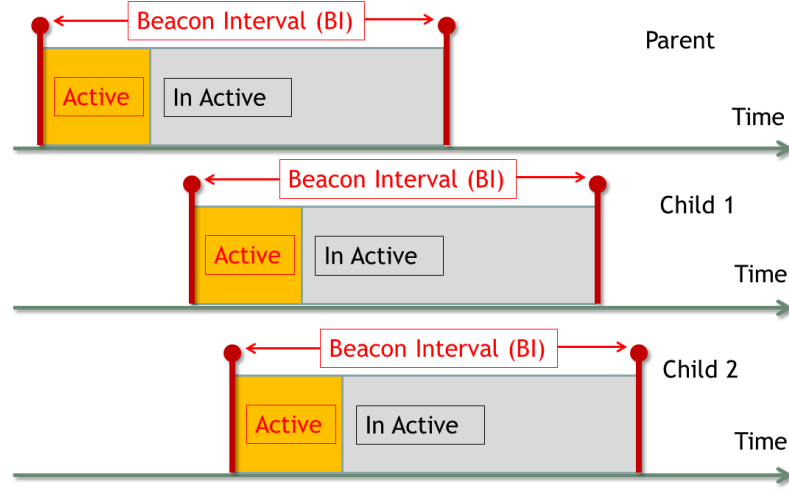


Fig. A.1: Beacon Separation in Time Domain to Avoid Collisions.

topology, the sensor nodes synchronize with their selected parent first and then start broadcasting periodic beacons. Therefore, in order to avoid collisions, the sensor node starts beaconing after it has received a beacon from its parent. That is, the beacon transmission of a sensor node occurs with respect to its parent's beacon reception. The desired effect is graphically shown in Figure A.1.

Figure A.1, shows a delay between the transmission of parent's beacon and the transmission of children's beacon. In addition, Figure A.1 also shows a delay between the transmission of beacons between two children devices.

The EMMON project (Tennina et al., 2011a,b) used time slot negotiation technique to achieve the desired effect and it schedules beacons to avoid collisions, where the child device requests the beacon start-time offset from the parent device after the child has associated with the parent. In order to achieve the desired effect in this thesis, Equation (A.1) is used to obtain the starting time of periodic beacon broadcasts with respect to the parent device, while avoiding additional control overhead.

$$\begin{aligned}
\text{BeaconStartTime} = & \text{SuperframeDuration} \\
& + (\text{DeviceAddress mod DevicesPerLevel}) \\
& \times \text{BaseSuperframeDuration}
\end{aligned} \tag{A.1}$$

The separation between parent's beacon reception and child's beacon transmission is achieved by adding *SuperframeDuration* to the resultant starting time. The *SuperframeDuration* parameter is considered to be constant for a given network topology, and is configured at deployment time. The *SuperframeDuration* parameter reflects the active period, during which a device listens after transmitting its beacon.

The separation between beacon transmissions of sibling devices is achieved by adding a unique number of *BaseSuperframeDuration* periods to the resultant starting time. The *BaseSuperframeDuration* is a constant, set at 960 symbols, and is considered to be the unit of time for the implementation. The *SuperframeDuration* is also composed of a number of *BaseSuperframeDuration* periods, and is calculated by $\text{SuperframeDuration} = \text{BaseSuperframeDuration} \times 2^{SO}$, where SO is Superframe Order and is a configurable parameter.

Equation (A.1) shows that the unique number of *BaseSuperframeDuration* periods to be added to the starting time of a device's periodic beaconing is based on the device address and the maximum number of devices per level in the network topology. Here, the device address is assumed to be unique throughout the network, which is a very common assumption in many networks.

This mechanism ensures that unique starting times are obtained based on unique device addresses. Indirectly, it also ensures that the beacons from any two devices do not collide as long as the number of devices within a single communication radius are low enough to accommodate their beacons within the beacon interval of the root device (Fixed Gateway). The devices that are not in communication range with the root device, are not effected by this limitation. Figure A.2 shows that devices within the blue circle will be effected by this limitation. This means that the number of devices within the

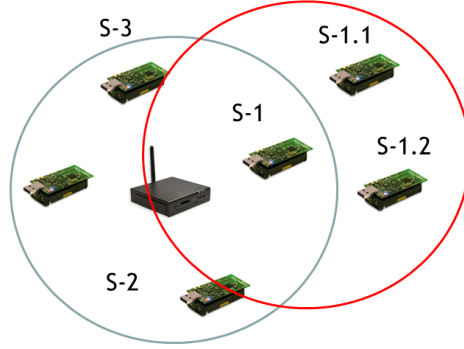


Fig. A.2: Spatial Limitation on Beaconing Devices.

central (blue) circle should be low enough to accommodate their beacons with the beacon interval of the fixed gateway. However, the child devices of sensor node S1 (Figure A.2) do not depend on the beacon interval of the fixed gateway. However, the child devices of sensor node S1 will be effected by similar limitation which will be based on the local root device (S1 in this case) within their communication radius.

A.2 Finding possible children of a given node

In order to find children nodes for each of the given node (nodes present in the applicable node list), the mobile device infers the children of the given node from the set of sensor nodes that appeared in its scan (nodes in range).

While searching for children of a given node, the notion of “Possible Children” is used instead of “Absolute Children” because due to the presence of mobile devices, the network topology may get inverted, i.e., higher order nodes (e.g., cluster heads) may forward traffic via lower order node (e.g., end device), due to its improved performance in presence of mobile devices. For this reason, the children of a given node are identified based on its credentials rather than encapsulating the unique address of the parent node in the beacons of the children nodes that can essentially identify absolute children of any given node.

Since the credentials of the given sensor node are known and each sensor node calculates its credentials based on the credentials of its next hop node using Equation (3.14), which is then modulated based on the risk of losing packets while forwarding via a particular device type using Equation (3.17), the mobile device can infer possible children of the given node using Equation (3.14) as the degradation factor of the credentials per hop is assumed to be known. Due to the risk of packet loss, the mobile device calculates a range in which the expected credentials of the children node of a given node may fall.

For example, if the expected credentials of a child are calculated as $Cr'(Child(n)) = Cr(n) \times (1 - \xi)$, then the lower bound of these expected credentials is calculated as $Cr'(Child(n)) - Cr'(Child(n)) \times \zeta$, and the upper bound of the expected credentials is calculated as $Cr'(Child(n)) + Cr'(Child(n)) \times \zeta$, where for a 10% error bound, $\zeta = 0.10$.

The mobile device then searches for nodes that advertised their credentials within these bounds, and form a subset of DS (set of discovered sensor nodes), say $DS' \subseteq DS$. This subset holds only those sensor nodes that can potentially be a child of the given node. The mobile device then computes a subset of DS' with only those nodes that are within radio range of the given node, such that $DS'' \subseteq DS'$.

This assumes that the location of each sensor node is also broadcasted in the beacon frame and the radio range of sensor nodes is known a priori. Because, the transmission power of sensor nodes is considered to be configured by the network administrator, according to the requirements associated with network lifetime, knowing the radio transmission range of sensor nodes is a reasonable assumption to make.

Thus, the set DS'' will hold all nodes that can potentially be children of the given node (or the nodes that may be forwarding using the given node).

Appendix B

Basic Communication Components

B.1 Communication between more capable devices

Communication between more capable devices is established using the peer-to-peer TCP/IP client server communication model, such that devices on both ends act as a server and a client at the same time enabling any device to initiate communication with the other, provided the IP address is known. This paradigm is applicable to the communication between the centralized command and control server (CC) and the more capable devices co-located with WSN, such as the fixed gateway and the mobile devices.

At the start of the network, when either a fixed gateway or a mobile device becomes available, it establishes a connection with the CC¹, and registers itself (containing IP address and geographic location). From this point onward, if the CC needs to send an unsolicited message, e.g., initiate geographic sensor data queries, to the fixed gateway or any other mobile device, it can establish a connection with the targeted device, as the IP address will be known. This, however, assumes that the IP address of the target device

¹Assuming the public IP address or URL of the CC is known a priori.

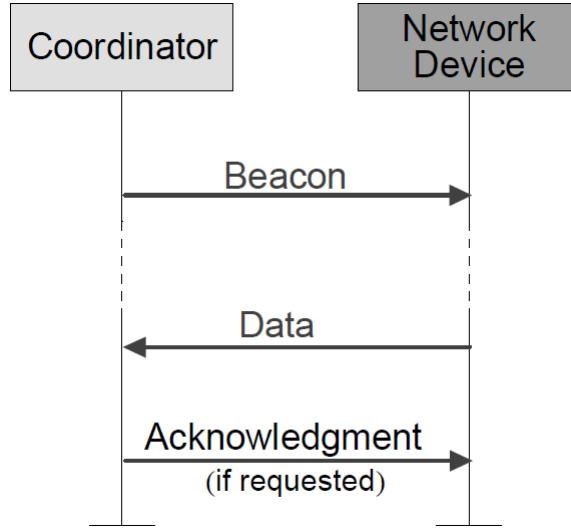


Fig. B.1: Data transfer to a parent node (Coordinator) from a child node (Network Device), IEEE802.15.4 (2006).

has not changed since the last registration request was received at the CC.

B.2 Communication between sensor nodes

In this thesis the sensor network is assumed to be deployed in a hierarchical cluster-tree topology (Koubaa et al., 2006; Tennina et al., 2011a,b), where the root node is considered to be the fixed gateway within a PAN communicating over a single radio channel. In this topology, any node can communicate directly with either their parent or children but not with a sibling node.

The following sections discuss mechanisms involved to establish upward (from a child node to a parent node) and downward (from a parent node to a child node) communication respectively.

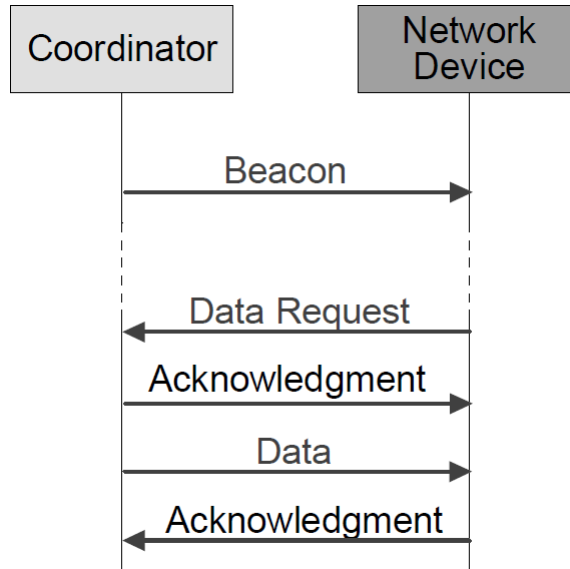


Fig. B.2: Data transfer to a child node (Network Device) from the parent node (Coordinator), IEEE802.15.4 (2006).

B.2.1 Upward communication

In the upward communication flow, the data originates at the sensor nodes and travels towards the fixed gateway via multi-hop paths.

After synchronizing with a parent, a sensor node transmits a packet to its parent after receiving a beacon from the parent during the active portion of the superframe of the parent (IEEE802.15.4, 2006) (Figure B.1). Using a mechanism similar to beacon collision avoidance (Appendix A, Section A.1), each node schedules its transmission to the parent according to its address.

B.2.2 Downward communication

In the downward communication flow, the data packets travel in the opposite direction of the fixed gateway, i.e., from parent to child node.

As can be seen in Figure B.2, the downward communication mechanism available in

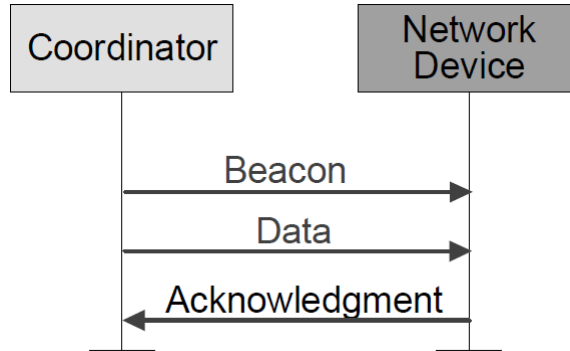


Fig. B.3: Data transfer to a child node (Network Device) from the parent node (Coordinator), as a broadcast packet with addressing fields in the network layer header.

IEEE802.15.4 (2006) beacon-enabled PAN (indirect transmission mode) requires a few data exchanges for the transmission to complete, such that if a packet is available for a particular child node at the parent, then an indication is provided in the beacon frame of the parent node. The child node, thus, has to request the packet from the parent upon realizing the availability of the packet. This transaction must complete within the active portion of the parent node's superframe.

Due to considerations on energy consumption and data request collisions from multiple children, downward communication flow was modified in this research, such that the children nodes keep their radio in listening mode after receiving the beacon from the parent node. The parent node broadcasts the first data packet in its queue, after broadcasting the beacon frame (Figure B.3), containing addressing fields in the network layer header. Except for the addressee, all other children nodes reject the packet. This, however, reduces the throughput of downward packet transmissions, as only a single packet may be communicated to the children nodes in a single beacon interval, trading off latency in favour of energy conservation and collisions. Although, multiple payloads to the same destination can be combined together in a single packet, e.g., if multiple sensor readings originating from different nodes are to be sent to the same mobile device.

Another alternative is to utilize guaranteed time slots (GTS) (IEEE802.15.4, 2006),

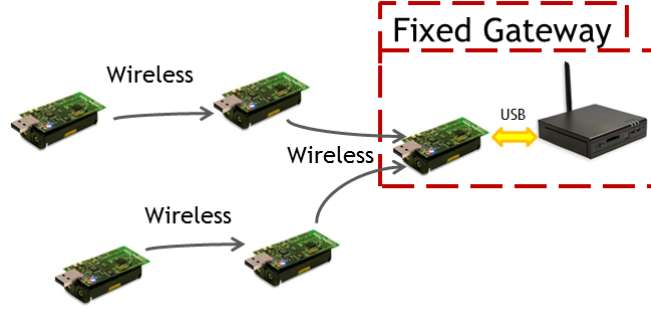


Fig. B.4: Fixed gateway (host) connected with a sensor node over USB communicates with other sensor nodes over wireless link.

however, it increases the memory consumption (ROM), especially in memory-constrained sensor nodes such as TelosB.

B.3 Communication between more capable devices and sensor nodes

With commercially-available off-the-shelf components, integrating more capable mobile devices in urban-scale WSN becomes particularly difficult as the sensor nodes and mobile devices are incompatible with each other due to the different classes of radio modules available on each. Commonly, sensor nodes communicate over low power IEEE802.15.4 (2006) radios, whereas the more capable devices use WiFi (IEEE 802.11) and Bluetooth (IEEE 802.15.1) radios. For the sake of simplicity in this research, laptop computers function as mobile devices and fixed gateways. They are connected to a sensor node, e.g., Crossbow (2005)², over USB (Figure B.4), which acts as a technology bridge enabling communication with the rest of the WSN.

²TelosB can communicate over IEEE802.15.4 (2006).

B.4 Energy consumption on mobile devices with respect to different radio technologies - A personal experience

Personal usage observations of the author of this thesis suggest that when only GSM is operational, the battery can last more than a day on a relatively new smart phone, however, if WiFi is enabled throughout the day, then the battery on the same phone will last approximately 15 to 16 hours, depending on the usage and characteristics of the individual applications. However, if 3G/4G radio is enabled, then the battery will hardly last for 12 hours. The energy depletion patterns also depend on the capacitance of the battery, its manufacturing process and its overall life, i.e., how long it has been in use. It can be observed that the battery life time on old phones is significantly less than that of a new phone, even if the make and model are the same, meaning that battery life gets shorter with the ageing of the battery.

Appendix C

Algorithms

C.1 Sensor node selection at mobile device

When a mobile device discovers sensor nodes in its vicinity, it selects one of those sensor nodes to fulfill the prioritized objectives depending on the objective's policy.

Algorithm 1 Sensor node selection depending on the objective using the Node Selection Context.

```
1: function GETNODESELECTIONCONTEXT.SELECT(SensorNodesInVicinity, Objective)
2:   PreSelectionFilter  $\leftarrow$  GetPreSelectionFilter(Objective)
3:   SelectionStrategy  $\leftarrow$  GetSelectionStrategy(Objective)
4:   PostSelectionProcessor  $\leftarrow$  GetPostSelectionProcessor(Objective)
5:   if PreSelectionFilter  $\neq$  NULL then
6:     ApplicableNodes  $\leftarrow$  PreSelectionFilter.filter(SensorNodesInVicinity)
7:   else
8:     ApplicableNodes  $\leftarrow$  SensorNodesInVicinity
9:   end if
10:  SelectedNode  $\leftarrow$  SelectionStrategy.select(ApplicableNodes, SensorNodesInVicinity)
11:  if PostSelectionProcessor  $\neq$  NULL then
12:    SelectedNode  $\leftarrow$  PostSelectionProcessor.process(SelectedNode)
13:  end if
14: end function
```

Algorithm 2 Creation of pre-selection filter.

```
1: function GETPRESELECTIONFILTER(Objective)
2:   if Objective = FaultTolerance then
3:     return NULL
4:   else
5:     if Objective = EnergyConservation then
6:       return LowEnergyFilter
7:     else
8:       if Objective = CongestionReduction then
9:         return HighTransmissionInLastMinuteFilter
10:      else
11:        return NULL
12:      end if
13:    end if
14:  end if
15: end function
```

Algorithm 3 Creation of the selection strategy.

```
1: function GETSELECTIONSTRATEGY(Objective)
2:   if Objective = EnergyConservation OR Objective = CongestionReduction
   then
3:     return TrafficReductionStrategy
4:   else
5:     return GreedyCredentialImprovementStrategy
6:   end if
7: end function
```

Algorithm 4 Creation of post-selection processor.

```
1: function GETPOSTSELECTIONPROCESSOR(Objective)
2:   if Objective = FaultTolerance then
3:     return LowCredentialPostProcessor
4:   else
5:     return NULL
6:   end if
7: end function
```

Appendix D

Simulation

The following sections discuss a few implementation details incorporated in the simulator for the evaluation of the STERIOD protocol.

D.1 Simulated node hierarchy

This simulator defined separate abstractions for each type of device present in the network, e.g., command and control server, fixed gateway, mobile devices and sensor nodes (Figure D.1). The object-oriented representations of the individual devices implemented a common abstract “Node” class that encapsulated the common functionality on all the devices. The “Node” class implemented simulated communication, in addition to maintaining common statistics of the concrete device types, e.g., the number of packets transmitted, received, lost etc. The concrete implementations of each of the device types included the specific behaviour. For example, the “End Device” class was responsible for generating sensor data packets, whereas, the command and control server was responsible to collect the data packets and update the statistics, e.g., average number of hops traversed by the data packets.

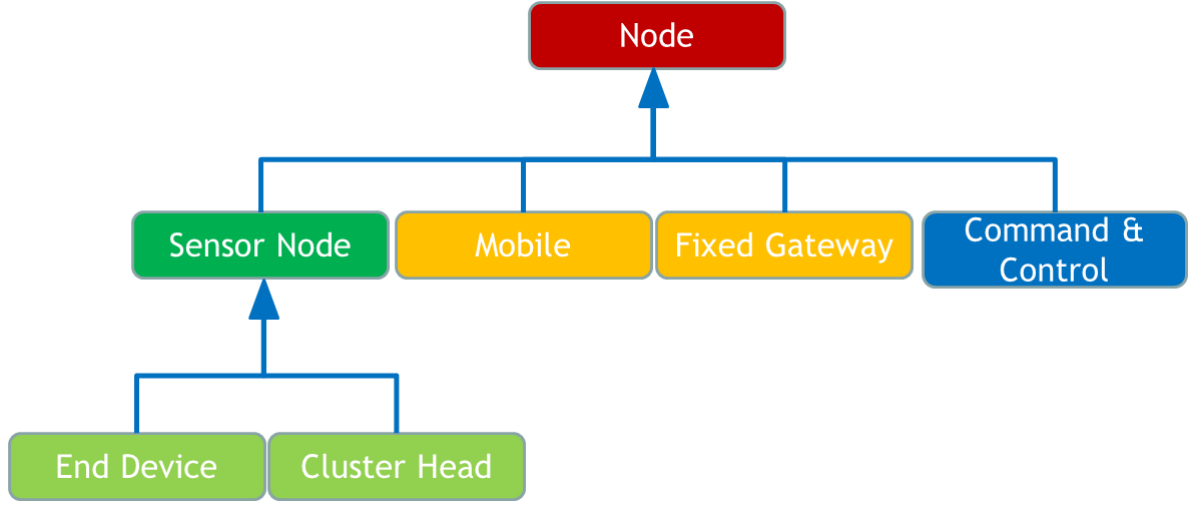


Fig. D.1: Representing different types of nodes in simulated environment.

D.2 Simulating mobility and location interpolation

When the initial and final locations are identified by the concrete location generator, either from a file or randomly, the distance to be covered is calculated using Equation (D.1). The distance to be covered is considered as the hypotenuse of an imaginary right-angled triangle, such that base and perpendiculars are given by Δx and Δy respectively. Using the three sides of this imaginary right-angled triangle, $\cos\theta$ and $\sin\theta$ are calculated using Equations (D.2) and (D.3). When an intermediary location is to be calculated, the values of $\cos\theta$ and $\sin\theta$ are used to calculate the current x and y location on the cartesian plane according to Equations (D.5) and (D.6), respectively, where the distance covered since the last location request was made, is calculated using Equation (D.4).

$$DistanceToBeCovered = \sqrt{\Delta x^2 + \Delta y^2} \quad (D.1)$$

$$\cos\theta = \frac{|\Delta x|}{DistanceToBeCovered} \quad (D.2)$$

$$\sin\theta = \frac{|\Delta y|}{DistanceToBeCovered} \quad (D.3)$$

$$\Delta DistanceCovered = \Delta Time * Speed \quad (D.4)$$

$$CurrentX = \begin{cases} PreviousX + \Delta DistanceCovered * \cos\theta & \text{if } FinalX > PreviousX \\ PreviousX - \Delta DistanceCovered * \cos\theta & \text{if } FinalX < PreviousX \end{cases} \quad (D.5)$$

$$CurrentY = \begin{cases} PreviousY + \Delta DistanceCovered * \sin\theta & \text{if } FinalY > PreviousY \\ PreviousY - \Delta DistanceCovered * \sin\theta & \text{if } FinalY < PreviousY \end{cases} \quad (D.6)$$

Glossary

AA This refers to standardized battery size. 150

ACK Acknowledgement. 78, 80, 81

ADV Advertisement packet. 109, 112, 142, 148, 158

ALBA-R Adaptive Load Balancing Algorithm with Rainbow protocol. 59

AM Amplitude Modulation. 166

AODV Ad hoc On-Demand Distance Vector routing protocol. 33–35, 52, 75, 81

API Application Program Interface. 9, 16

BI Beacon Interval. 25, 106, 107, 111, 112, 165

BO Beacon Order. xiv, 106, 108, 111–113, 142–144, 147–149, 151, 153, 159

BSD Base Superframe Duration. 107

CAP Contention Access Period. 126

CC Command and Control Server. 15–17, 39, 227, 228

CIA Cluster ID Announcement. 65, 66

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 30, 222

CTP Collection Tree Protocol. 41–43

CTS Clear To Send. 17, 60, 80, 81

dBm decibel-milliwatts: power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW). 112

DEAR Device and Energy Aware Routing protocol. 76, 83

DELAR Device-Energy-Load Aware Relaying (routing) protocol. 79–83, 168, 202, 204, 205, 211

DSDV Distance Sequenced Distance Vector routing protocol. 75, 79, 81, 205

DSR Dynamic Source Routing protocol. 28, 33–35, 52

DT Discovery Time period. 25

DTN Delay Tolerant Network. 14

EAR Energy Aware Routing. 205

EARM Energy Aware Routing for Mobile gateway in WSN. 35, 51, 63

EMMON EMbedded MONitoring, an ARTEMIS/EU Project on large-scale sensor networks. 42, 113, 136, 223

FM Frequency Modulation. 166

GEAR Geographical and Energy Aware Routing protocol. 36, 40

GPS Global Positioning System. 17, 141, 162, 197, 198, 219

GPSR Greedy Perimeter Stateless Routing protocol. 36, 40, 60

GSM Global System for Mobile communication. 166, 197, 232

GTS Guaranteed Time Slots. 126, 230

GW Gateway or Fixed Gateway. 6

HCB Hierarchical Cluster based Routing. 66, 67, 69

HOLSR Hierarchical Optimized Link State Routing. 64, 66, 69

HTC Hierarchical Topology Control. 65, 66

HTTP HyperText Transfer Protocol. 220

IAR Intelligent Agent-based Routing. 52, 63

ID Identifier. 38, 70

ILAD Inverse Location-Aware Discovery. 2, 10, 14, 18–20, 125, 136, 139, 148, 149, 152, 156, 159, 161–163, 166, 197, 198, 208, 212, 214, 217, 218

IP Internet Protocol. 16, 22, 39, 56, 61, 100, 124, 185, 219, 227

J2ME Java Mobile Edition. 137

JVM Java Virtual Machine. 137

LPP Low Power Probing. 27, 31

LR Long Range. 26

MAC Medium Access Control layer protocol. 79, 82, 85–87, 135, 137, 222

MANET Mobile Ad hoc Network. vii, 14, 22, 33, 51

MC-Routing Multi-Class Routing protocol. 78, 79, 82, 83

MDC Mobile Data Collector. xvi, 24, 25, 28, 29, 105–113, 139, 141–143, 145–153

MEMOSEN Multi-radio and Mobile Enabled Wireless Sensor Network. 61

MPR Multi-Point Relay. 65

MTU Maximum Transmission Unit. 73

MULE Mobile Ubiquitous LAN Extension. 25

nesC Network Embedded System C. 137

OLSR Optimized Link State Routing. 64, 65

OSI Open Systems Interconnection. 17

PAN Personal Area Network. 100, 102, 169, 228, 230

PC Personal Computer. 6

PGR Probabilistic Geographic Routing protocol. xv, 37, 38, 40

PRoPHET Probabilistic Routing Protocol using History of Encounters and Transitivity. 70, 72–74

QoS Quality of Service. 56, 63

RAM Random Access Memory. 137, 138

RErr Route Error. 33

REST Representational State Transfer. 219

RL Re-inforcement Learning. 93, 96, 97, 99

ROI Return On Investment. 8, 9

ROM Read Only Memory. 137, 138, 231

ROME Routing Over Mobile Elements protocol. 59, 62, 63

RREP Route Reply. 33

RREQ Route Request. 33

RSSI Radio Signal Strength Indicator. 56

RTS Request To Send. 17, 60

SCAR Sensor Context Aware Routing. 72–74

SEAD Scalable Energy-efficient Asynchronous Dissemination protocol. xv, 47–51, 63

SimBet Similarity and Betweenness routing. 71, 73, 74

SN Sensor Node. x, xvi, 25–27, 29, 105, 107–109, 111, 112, 141, 142, 145–153

SO Superframe Order. xvi, 107, 111, 113, 142–144, 150–153, 224

SOAP Simple Object Access Protocol. 219

SR Short Range. 26

STERIOD Smart Traffic Energy and Resource aware Offloading using Inverse Discovery. xvi, 2, 10, 14, 18–20, 32, 40, 41, 74, 136, 139, 140, 154, 155, 162, 163, 166–168, 170, 171, 179, 196–198, 204, 205, 207–209, 211, 214, 215, 218, 220, 222, 237

TC Topology Control. 66

TCP Transmission Control Protocol. 100, 124, 219, 227

TTDD Two Tier Data Dissemination protocol. xv, 44–46, 48, 50, 51, 53, 55, 63

TTL Time To Live. 70

URL Uniform Resource Locator. 16, 227

USB Universal Serial Bus. 231

WMEWMA Window Mean Exponentially Weighted Moving Average. 43

WSAN Wireless Sensor and Actuator Network. 3

WSN Wireless Sensor Network. vi, vii, ix, xiii, xv, 1–6, 8, 11, 13–16, 20–24, 26, 28, 31–34, 38, 39, 43–48, 50–53, 55, 56, 59, 61, 63, 73, 84, 88, 89, 91, 100, 103, 105, 106, 113, 136, 137, 153, 160, 163, 168, 185, 204, 207, 212–214, 216–218, 220, 227, 231

Bibliography

- Adjih, C., Baccelli, E., and Jacquet, P. (2004). Link state routing in wireless ad-hoc networks. In *Military Communications Conference, 2003. MILCOM 2003. IEEE*, volume 2, pages 1274–1279. IEEE.
- Ahmed, W., Huggard, M., and Mc Goldrick, C. (2009). An application adaptive energy model for wireless sensor nodes. In *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 147–150. ACM.
- Akkaya, K. and Younis, M. (2004). Energy-aware routing to a mobile gateway in wireless sensor networks. In *Global Telecommunications Conference Workshops (GlobeCom Workshops) 2004*, pages 16–21. IEEE.
- Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568.
- Ansari, J., Pankin, D., and Mähönen, P. (2009). Radio-triggered wake-ups with addressing capabilities for extremely low power sensor network applications. *International Journal of Wireless Information Networks*, 16(3):118–130.
- Avudainayagam, A., Lou, W., and Fang, Y. (2003). DEAR: A Device and Energy Aware Routing protocol for heterogeneous ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(2):228–236.

- Balasubramanian, N., Balasubramanian, A., and Venkataramani, A. (2009). Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 280–293. ACM.
- Basagni, S., Nati, M., Petrioli, C., and Petroccia, R. (2010). Rome: Routing over mobile elements in wsns. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–7. IEEE.
- Biswas, S. and Morris, R. (2005). ExOR: opportunistic multi-hop routing for wireless networks. *Computer Communication Review*, 35(4):133.
- Boyan, J. and Littman, M. (1994). Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in neural information processing systems*, pages 671–678.
- Buettner, M., Yee, G. V., Anderson, E., and Han, R. (2006). X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320. ACM.
- Burke, J. A., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., and Srivastava, M. B. (2006). Participatory sensing.
- Caballero, F., Merino, L., Gil, P., Maza, I., and Ollero, A. (2008). A probabilistic framework for entire wsn localization using a mobile robot. *Robotics and Autonomous Systems*, 56(10):798–806.
- Casari, P., Nati, M., Petrioli, C., and Zorzi, M. (2007). Efficient non-planar routing around dead ends in sparse topologies using random forwarding. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 3122–3129. IEEE.
- Chen, C. and Ma, J. (2006). MEMOSEN: multi-radio enabled mobile wireless sensor

- network. In *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, volume 2, page 5. IEEE.
- Chipcon, A. (2006). Cc2420 datasheet.
- Chu, D., Deshpande, A., Hellerstein, J., and Hong, W. (2006). Approximate data collection in sensor networks using probabilistic models. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 48–48. IEEE.
- Crossbow (2005). Memsic Corporation's WSN module TelosB, <http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html>.
- Daly, E. and Haahr, M. (2007). Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM.
- Di Francesco, M., Shah, K., Kumar, M., and Anastasi, G. (2010). An adaptive strategy for energy-efficient data collection in sparse wireless sensor networks. *Wireless Sensor Networks*, pages 322–337.
- Dohler, M., Watteyne, T., Barthel, D., Valois, F., Lu, J.-L., et al. (2007). Kumars, zipfs and other laws: How to structure an optimum large-scale wireless (sensor) network. In *13th European Wireless Conference*.
- Dowling, J., Curran, E., Cunningham, R., and Cahill, V. (2005). Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(3):360–372.
- Du, X., Wu, D., Liu, W., and Fang, Y. (2006). Multiclass routing and medium access control for heterogeneous mobile ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 55(1):270–277.

- Dunkels, A., Osterlind, F., Tsiftes, N., and He, Z. (2007). Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 28–32. ACM.
- Dyo, V. and Mascolo, C. (2007). A node discovery service for partially mobile sensor networks. In *Proceedings of the 2nd international workshop on Middleware for sensor networks*, pages 13–18. ACM.
- Dyo, V. and Mascolo, C. (2008). Efficient node discovery in mobile wireless sensor networks. *Distributed Computing in Sensor Systems*, pages 478–485.
- Forstert, A. and Murphy, A. (2008). FROMS: Feedback routing for optimizing multiple sinks in WSN with reinforcement learning. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 371–376. IEEE.
- Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., and Levis, P. (2009). Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14. ACM.
- Graefenstein, J., Albert, A., Biber, P., and Schilling, A. (2009). Wireless node localization based on rssi using a rotating antenna on a mobile robot. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, pages 253–259. IEEE.
- Gu, L. and Stankovic, J. (2005). Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems*, 29(2):157–182.
- He, T., Stankovic, J., Lu, C., and Abdelzaher, T. (2003). SPEED: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*.

- IEEE802.15.4 (2006). Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPANs).
- Intanagonwiwat, C., Govindan, R., and Estrin, D. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM.
- Jain, S., Shah, R., Brunette, W., Borriello, G., and Roy, S. (2006). Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 11(3):327–339.
- Johnson, D. and Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181.
- Karp, B. and Kung, H. (2000). GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM.
- Kim, H., Abdelzaher, T., and Kwon, W. (2003). Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 193–204. ACM.
- Kim, J., In, J., Hur, K., Kim, J., and Eom, D. (2010). An intelligent agent-based routing structure for mobile sinks in WSNs. *Consumer Electronics, IEEE Transactions on*, 56(4):2310–2316.
- Kimura, N. and Latifi, S. (2005). A survey on data compression in wireless sensor networks. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, pages 8–13. IEEE.
- Kohvakka, M., Suhonen, J., Kuorilehto, M., Kaseva, V., Hännikäinen, M., and

- Hämäläinen, T. (2009). Energy-efficient neighbor discovery protocol for mobile wireless sensor networks. *Ad Hoc Networks*, 7(1):24–41.
- Kondepu, K., Anastasi, G., and Conti, M. (2011). Dual-beacon mobile-node discovery in sparse wireless sensor networks. In *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC 2011), Corfu, Greece*.
- Koubaa, A., Alves, M., and Tovar, E. (2006). Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In *Real-Time Systems Symposium, 2006. RTSS'06. 27th IEEE International*, pages 412–421. IEEE.
- Kusy, B., Lee, H., Wicke, M., Milosavljevic, N., and Guibas, L. (2009). Predictive qos routing to mobile sinks in wireless sensor networks. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 109–120. IEEE.
- Lane, N. D., Eisenman, S. B., Musolesi, M., Miluzzo, E., and Campbell, A. T. (2008). Urban sensing systems: opportunistic or participatory? In *Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 11–16. ACM.
- Levin, R., Cohen, E., Corwin, W., Pollack, F., and Wulf, W. (1975). Policy/mechanism separation in Hydra. In *ACM SIGOPS Operating Systems Review*, volume 9, pages 132–140. ACM.
- Levis, P. and Gay, D. (2009). *TinyOS programming*. Cambridge University Press.
- Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., et al. (2005). Tinyos: An operating system for sensor networks. *Ambient intelligence*, 35.
- Liang, C.-J. M., Terzis, A., et al. (2008). Koala: Ultra-low power data retrieval in wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 421–432. IEEE Computer Society.

- Lindgren, A., Doria, A., and Schelén, O. (2004). Probabilistic routing in intermittently connected networks. *Service Assurance with Partial and Intermittent Resources*, pages 239–254.
- Liu, W., Zhang, C., Yao, G., and Fang, Y. (2011). Delar: A device-energy-load aware relaying framework for heterogeneous mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 29(8):1572–1584.
- Lu, G., Krishnamachari, B., and Raghavendra, C. (2004). Performance evaluation of the ieee 802.15. 4 mac for low-rate low-power wireless networks. In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 701–706. IEEE.
- Marcelloni, F. and Vecchio, M. (2008). A simple algorithm for data compression in wireless sensor networks. *Communications Letters, IEEE*, 12(6):411–413.
- Mohamed, M., Wu, W., and Moniri, M. (2013). Adaptive data compression for energy harvesting wireless sensor nodes. In *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, pages 633–638. IEEE.
- Mohan, M., Bhattacharya, A., Bouroche, M., and Cahill, V. (2012). Bless a divine energy saving protocol for wireless sensor networks. *Wireless days*.
- Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., and Boda, P. (2009). Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 55–68. ACM.
- Pasztor, B., Musolesi, M., and Mascolo, C. (2007). Opportunistic mobile sensor data collection with scar. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pages 1–12. IEEE.

- Perkins, C. and Royer, E. (1999). Ad-hoc On-Demand Distance Vector Routing, in proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications. *New Orleans, LA*, pages 90–100.
- Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM Computer Communication Review*, volume 24, pages 234–244. ACM.
- Pottie, G. J. and Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58.
- Priyantha, N. B., Balakrishnan, H., Demaine, E. D., and Teller, S. (2005). Mobile-assisted localization in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 172–183. IEEE.
- Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M. B. (2002). Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2):40–50.
- Roosta, T. and Menzo, M. (2004). Probabilistic geographic routing protocol for ad hoc and sensor networks. *Department of EECS UC Berkeley*.
- Sadaphal, V. and Jain, B. (2008). Random and periodic sleep schedules for target detection in sensor networks. *Journal of Computer Science and Technology*, 23(3):343–354.
- Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M. (2002). Optimizing sensor networks in the energy-latency-density design space. *Mobile Computing, IEEE Transactions on*, 1(1):70–80.
- Somasundara, A., Kansal, A., Jea, D., Estrin, D., and Srivastava, M. (2006). Controllably mobile infrastructure for low energy embedded networks. *Mobile Computing, IEEE Transactions on*, 5(8):958–973.

- SunSpot (2004). Sun Microsystem's Java based WSN Platform, <http://www.sunspotworld.com/>.
- Sutton, R. and Barto, A. (1998). *Reinforcement learning: An introduction*. The MIT press.
- Tennina, S., Bouroche, M., Braga, P., Gomes, R., Alves, M., Mirza, F., Ciriello, V., Carrozza, G., Oliveira, P., and Cahill, V. (2011a). Emmon: A wsn system architecture for large scale and dense real-time embedded monitoring. In *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on*, pages 150–157. IEEE.
- Tennina, S., Gomes, R., Alves, M., Ciriello, V., and Carrozza, G. (2011b). The dark side of demmon: what is behind the scene in engineering large-scale wireless sensor networks. In *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 41–50. ACM.
- Tesauro, G. (2007). Reinforcement learning in autonomic computing: A manifesto and case studies. *IEEE Internet Computing*, pages 22–30.
- Vahdat, A. and Becker, D. (2000). Epidemic routing for partially connected ad hoc networks. Technical report, Citeseer.
- Verdone, R., Dardari, D., Mazzini, G., and Conti, A. (2010). *Wireless sensor and actuator networks: technologies, analysis and design*. Academic Press.
- Villasenor-Gonzalez, L., Ge, Y., and Lament, L. (2005). HOLSR: a hierarchical proactive routing mechanism for mobile ad hoc networks. *Communications Magazine, IEEE*, 43(7):118–125.
- Vlissides, J., Helm, R., Johnson, R., and Gamma, E. (1995). Design patterns: Elements of reusable object-oriented software. *Reading: Addison-Wesley*, 49:120.

- Wan, C.-Y., Eisenman, S. B., and Campbell, A. T. (2003). Coda: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 266–279. ACM.
- Wang, H., Zhang, X., Naït-Abdesselam, F., and Khokhar, A. (2007). Dps-mac: An asynchronous mac protocol for wireless sensor networks. In *High Performance Computing—HiPC 2007*, pages 393–404. Springer.
- Woo, A., Tong, T., and Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27. ACM.
- Xia, Y., Yeo, C., and Lee, B. (2009). Hierarchical Cluster Based Routing for Highly Mobile Heterogeneous MANET. *School of Computer Engineering, Nanyang Technological University, Singapore*.
- Ye, F., Luo, H., Cheng, J., Lu, S., and Zhang, L. (2002). A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, page 159. ACM.
- Yu, Y., Govindan, R., and Estrin, D. (2001). Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. *UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023*.