PHD

Wireless Sensor Network Routing Protocols for Data Aggregation

Pourroostaei Ardakani, Saeid

Award date:
2014

Awarding institution:
University of Bath

Link to publication

Wireless Sensor Network Routing Protocols for Data Aggregation

submitted by

Saeid Pourroostaei Ardakani

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Sciences

October 2014

COPYRIGHT

Signature of Author ....

Saeid Pourroostaei Ardakani

Abstract

Wireless Sensor Networks (WSNs) comprise a number of sensor nodes which typically sense, measure and report environmental data. The nodes themselves are highly resource constrained. They are typically networked in a self-organising manner without any specific infrastructure or centralised control. The key objective of WSN protocols is to minimise the cost of ambient data collection. Ambient data samples need to be collected and forwarded through minimum cost links (in terms of hop count and consumed energy) to data consumer access point (sink) for further analysis and manipulation.

WSN routing is the field of research that focuses on the interconnection of sensor nodes via either single or multi-hop paths to forward data packets from event regions to the sink. However, the routing overhead increases if raw data packets are forwarded from each source region to the sink. Data aggregation is therefore a technique that has the potential to collect and combine data packets to express the collected information in a summary form. It reduces the number and size of transmissions and eliminates redundant data packets. WSN Routing can be performed in two models for data aggregation: mobile agent and client/server. The former routes mobile agent(s) to collect and aggregate data samples from the sensor nodes, whereas the latter establishes an hierarchical network in which data packets are aggregated and forwarded from the ambient event regions to the sink in a convergent manner.

Data aggregation routing aims to maximise the number of collected data samples, while minimising energy consumption and data collection delay. Minimising energy consumption is a vital requirement due to resource constraints in WSNs. Data collection delay should be minimised as it is the key to data freshness. At the same time, the number of collected data samples should be maximised, as it should lead to increased accuracy and robustness in data collection.

To achieve the objectives of data aggregation routing, this thesis proposes two data aggregation protocols: one for mobile agents (called ZMA) and another using client/server (named CBA). ZMA uses multiple mobile agents to collect and aggregate over a WSN. The mobile agents start their journeys in a bottom-up manner from the event regions to the sink. They visit the sensor nodes and collect and aggregate data samples and then return to the sink to deliver the (aggregated) results. CBA partitions the network into a set of data-centric clusters and then establishes a spanning tree from the cluster-heads to the sink to forward and aggregate data packets hierarchically.

The performance of the proposed protocols is tested and evaluated through simulation. The simulation results of each of the proposed protocols are compared against two well-known protocols for each routing model, namely TBID (Tree-Based Itinerary Design) and NOID (The Near-Optimal Itinerary Design algorithm) for mobile agents and LEACH (Low-Energy Adaptive Clustering Hierarchy) and DDiFF (Directed Diffusion) for client/server. The results indicate that the proposed algorithms perform well compared to the respective benchmark protocols in most circumstances.

أميري حسين ونعم الأمير

Acknowledgements

"Human beings are members of a whole,
In creation of one essence and soul.

If one member is afflicted with pain,
Other members unease will remain.

If you have no sympathy for human pain,
The name of human you cannot retain."

Sa'di of Shiraz (1258 C.E.)

One of the joys of completion is to look over the journey past and remember all the people who have helped and supported me along this long but fulfilling road.

First and foremost, I wish to express my sincere gratitude to my advisors, Dr. Julian Padget and Dr. Marina De Vos, for insightful advices, conversations and suggestions during these past four years. I am very proud of what we have achieved together.

I would like also to thank my PhD committee, Prof. James Davenport (University of Bath) and Dr. Dritan Kaleshi (University of Bristol), for their valuable comments and suggestions, and of course hard questions.

I wish to take this opportunity to acknowledge the scholarships received towards my PhD, offered by Faculty of Science in University of Bath.

I will be thankful to my colleagues, Dr. Jim Laird, Dr. Fabio Nemets, Dr. Shadi Basurra, Dr Matthew England, Dr. Essra Alwan, Mrs. Fatemeh Jahedpari and Mr. Michael Wright, for all their helps and comments throughout. A very special thank to Dr. Craig Owen for his encouragements, interesting discussions and good wishes.

I am as ever, truly and deeply indebted to my parents, Mr. Hassan Pourroostaei Ardakani and Mrs. Tooran Khan Ardakani, for their great supports at every stage of my personal and academic life, and longed to see this achievement come true. Their unconditional love is always appreciated throughout my whole life.

I wish to pay high regards to the rest of my family especially Zohreh, Zahra, Elahe and Mohamad Reza, for their best wishes and encouragements which cheered me up to follow my dreams.

Last but not the least, I would like to express my great appreciation to my beloved queen, Dr. Zahra Khaleghi, without whose love, patience and understanding I would not have completed this work.

Dedicated to my parents
Baba Hassan and Maman Tooran

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless sensor network routing for data aggregation is the process of forwarding ambient data samples from the sensor nodes to the data consumer to deliver the results in an aggregated form. Data aggregation is a technique used in wireless sensor network routing to reduce size and number of data packets, and hence to increase network lifetime. The objective of data aggregation routing is to maximise the number of collected data samples at the same time as minimising network resource consumption and data collection delay. Maximising the number of reported data samples increases data collection accuracy and robustness. This means that increasing the number of captured data samples in the aggregation procedure should provide a precise summary of data samples to data consumers. It gives the data consumer the ability to make better decisions based on the collected data. Minimising the network energy consumption should result in increased network lifetime. Energy is a critical issue in wireless sensor networks as power is typically a very limited resource for the sensor nodes. Consequently, the network should be available for a longer period if the energy is used efficiently. Minimising delay is key to data accuracy and freshness in data aggregation routing. Data samples are expired or lose their meaning if they are delivered late to the sink.

This dissertation aims to enhance the performance of data aggregation routing in wireless sensor networks by achieving the above objectives. It proposes a protocols for each routing model, namely client/server and mobile agent, to (1) increase data collection accuracy by capturing as many data samples as possible, (2) increase network lifetime by reducing routing overhead and network energy consumption, and (3) reduce routing delay by establishing minimum hop count paths.

1.1 Wireless Sensor Network (WSN)

A *Wireless Sensor Network* (WSN) is an infrastructure comprised of a number of computing devices that have the ability to sense/measure and report ambient events. The objective of WSN deployment is to provide consumers the ability to sense/measure, monitor and/or react to specified environmental events. The term *consumer* is used throughout this dissertation to refer to the concept of data consumer where this does not create ambiguity. WSN consumer can be a civil, governmental, agricultural, commercial or industrial entity [Sohraby et al., 2007].

The sensor nodes usually are a set of inexpensive, small and/or tiny electronic devices that are able to perform three tasks: (1) measuring physical quantities (such as temperature or light) from the surrounding environment, (2) processing (and storing) the sensed/measured data (3) transmitting the information to a collection point of network called *Sink* or *Base Station* for either future processing or the consumer access. The sensor data are forwarded as wireless signal/packets through either single or multi-hop wireless communication links.

WSNs are becoming increasingly common due to low cost deployment and maintenance in comparison to other type of networks like ad-hoc. The sensor nodes are typically light, small and cheap. They are equipped by weak and basic communication, computation and storage modules which allow them to perform lightweight computing and communication tasks. WSNs do not need any specific infrastructure to be deployed. Indeed, they would be dynamically set up in an ad-hoc fashion and without any centralised control. For these reasons, WSN technology have attracted the interest of researchers as well as engineers. According to IDTechEx report [Harrop and Das, 2013], the market of WSN technology will grow from \$0.45 billion in 2011 to \$ 2 billion in 2021. Furthermore, a research and market report in 2013 [Harbor Research, 2013] forecasts that the market would grow rapidly and the Compound Annual Growth Rate (CAGR) would be anywhere between %55 to %130 over 2013 till 2016.

The concept of wireless sensor network is not particularly new and has existed in different forms for several decades. The history of WSN emergence goes back to 1950s when the United States deployed an extensive acoustic network called Sound Surveillance System (SOSUS) for submarine surveillance. It is still being used today by the National Oceanographic Atmospheric Administration (NOAA) to observe and monitor underwater earthquakes in the ocean. In the early 1980s, the major impetus for the research emerged when the United States Defence Advanced Research Projects Agency (DARPA) sponsored researchers to work on Distributed Sensor Nodes (DSN) project. DSN was a sensor network in which the nodes could communicate with each other using

Transmission Control Protocol/ Internet Protocol (TCP/IP) and Advanced Research Projects Agency Network (ARPAnets). The main goal of DSN research and deployment was measuring ambient data and then routing the results to the specific nodes for analysis and use. The key features of the project were: distributed computing, signal processing and object tracking. Carnegie Mellon University (CMU) researchers designed the network operating system named Accent [Rashid and Robertson, 1981] to enhance the network transparency and Massachusetts Institute of Technology (MIT) proposed a knowledge-based technique for signal processing (KBSP) [Milios, 1984]. Whilst the project took a long time to be built, there was ultimately no proper application for the DSN as it was not mature enough for the market. For example, the sensor nodes were very large (something like a shoe box) and the wireless connectivity patterns as well as standards were restricted.

The new wave of research on WSN started in the early 1990s focusing on networking techniques and information processing. During this era, the sensor nodes started to become smaller and cheaper. The deployment cost and time of new WSNs was reduced as they were deployed without requiring any specific communication infrastructure to be established. As a result, this technology initially attracted more international involvement and then more civilian and commercial applications (mainly environment monitoring) emerged around the world. Computing and communication advances in the late 1990s and early 2000s have led to a significant shift in this research area and pushed it closer to achieving the present vision of WSN. In 2001, DARPA offered again a new research programme called SensIT to provide a sensor network which had the ability to perform dynamic and infrastructure-less deployment, multitasking, dynamic querying and pre-programming. Various companies and industries funded the researchers to design and deploy this technology for novel applications. Based on the requirements to reduce the network deployment cost, the IEEE organisation defined IEEE 802.15 standards to offer low data rate wireless communication. Proceeding this, ZigBee Alliance proposed ZigBee standard for high level communication protocols. These standards were designed for low to very low duty cycle wireless networks. These standards result in (1) reducing wireless communications cost, (2) advances in Micro-Electro-Mechanical Systems (MEMSs) as well as Nanoscale Electro-Mechanical Systems (NEMS) to design tiny and cheap sensor nodes, (3) deploying dense, large and robust WSNs. Finally, further development and improvement of technologies, such as IEEE 802.15, WiMax and Bluetooth, have made WSN possible for such networks to be utilised for personal and commercial applications.

The remainder of this chapter is organised as follows: Section 1.2 briefly describes the benefits and possible application of Wireless Sensor Networks. Section 1.3 discusses wireless sensor network classifications and types in terms of their characteristics and distinctive features. Section 1.4 introduces sensor data collection and explains then data aggregation as a possible technique to enhance its performance in WSN. This section also discusses routing as one of the key components of WSN's data aggregation. Section 1.5 represents the research problems and questions (based on WSN's distinctive features) that are to be addressed in this thesis. Finally a brief outline of this research is provided in Section 1.6.

1.2 Benefits and Possible Applications of WSNs

The key benefit of WSNs is that they can be implemented almost anywhere without the need for any specific communication infrastructure. The sensor nodes are networked in a self-organising manner in many applications that require unattended operations. It allows a WSN to be deployed as an alternative to non-existent infrastructure (for cost effectiveness) or if the existing infrastructure is not appropriate to use. The following are seen as possible applications for WSN.

- **Military applications**: WSN can be a part of C4ISRT systems (Command, Control, Communications, Computing, Intelligence, Surveillance, Reconnaissance and Targeting) which can be utilised in military operations [Akyildiz et al., 2002]. Quick network deployment, self-organisation and fault tolerance characteristics of this technology attract researchers' attention to utilise WSNs to collect, monitor and report the environmental data that are interesting for military operations. Military applications are usually complex and critical. They use a set of multifunctional sensor nodes which have the capability to sense a broad range of ambient data. For example, monitoring friendly troops health and equipment is one of the WSN's application in military operations [Sohraby et al., 2007]. Small sensor nodes can be attached to the commanders, military equipment and ammunition to sense the conditions and report the data (through satellite for example) to a central computer that is accessible by the leaders. In turn, the leaders would be informed about the battlefield conditions and could therefore frequently organise the troops, equipment and ammunitions efficiently. Military object tracking and targeting, chemical/biological attack detections, battlefield surveillance and damage assessment can be other applications of this class.

- **Environmental applications**: they focus on collecting and reporting ambient data for the general consumers. The key applications of this class are habitat monitoring, forest fire detection, flood detection and precision agriculture. For example, Emergence ALERT System (EAS) [Agency, 2013] is a national warning system that was deployed by the USA in 1997 to detect floods. This system comprised of rainfall, water-level and weather sensors that sensed and reported the data to a central computer for further processing. This allows the system to detect a flood when the sensor nodes report the level of water is greater than a certain threshold. Animal habitat monitoring is another example of the applications in which the animal movement patterns, habitats and behaviours are monitored. The sensor nodes, which can be static or mobile (assigned to the animal bodies), collect the location information and then transfer the data to a central machine that is in charge of data collection. The data is analysed by the biology scientists to monitor/model the animal migration behaviours and/or nesting over different seasons [Mainwaring et al., 2002].

- **Health applications**: a number of WSN health applications have been designed in response to pre-hospital and in-hospital emergency care, ranging from disaster response to stroke patient rehabilitation. The key WSN health applications are remote monitoring of medical status, drug administration and medication intake monitoring; activities of daily living monitoring and elderly assistance. As an example, in resuscitative care, the sensor nodes continuously measure and integrate the vital signs like heart rate and transmit the results to the main computer to store in the patient care record. In this case, doctors and specialists would be able to observe the patient behaviours and reactions over either short or long periods and then provide required health care services based on the acquired information. As an example, a system (called Code-Blue) comprising of a small and wearable wireless pulse oximeter and two electrocardiogram (EKG) has been developed by Harvard university to measure heart-rate and oxygen saturation [Lab, 2013]. The EKGs send the integrated data to any other user end-point device like laptops or PDAs over a 100 meter wireless network. The system is programmed to process the vital sign data and then raise alerts when the signs fall outside the normal threshold. Thus, doctors would be able to observe the patients' status in real-time to provide required health-care services. A software named Code-Blue system software also is designed and developed to provide routing, naming, discovery and security services for the wireless medical sensors system.

- **Home applications**: WSN has the potential to provide monitoring, conservation, convenience and safety services at home. This technology can be utilised to establish home smart spaces by monitoring home resources as well as controlling equipment automatically and remotely. This technology would be beneficial for consumers as it facilitates (remote) home resource monitoring and harnesses intelligence to optimise resource consumption. For example, hotel HVAC (Heating, Ventilation and Air Conditioning) system (developed by SIMENS) uses a set of sensor nodes that are capable of monitoring the lighting, heating and cooling used in hotels [SIMENS, 2013]. This system monitors equipment and resources to optimise their utilisation at the buildings. HVAC sensors monitor the consumption of resources, check their current status and finally report the data to a central computer through a ZigBee based wireless communication infrastructure.

- **Commercial and industrial applications**: WSN can be utilised in industry to provide a set of advantageous services such as improving the equipment/resources management (frequent sensor monitoring), reducing the energy cost (optimising the manufacturing procedures), enhancing automatic systems (reducing user intervention and automating data collection) and extending existing manufacturing and process control systems. Environmental monitoring in offices, inventory control, vehicle tracking and detection and traffic flow surveillance are some of the key applications of this class. For example, the Traffic Pulse Technology system developed by Traffic.com [Traffic.com, 2013] collects traffic data including lane-by-lane travel speeds, lane occupancy and vehicle count, using a wireless sensor network and then aggregating and transmitting data to the central machines in a real-time manner. Using this system, the consumers are able to continuously monitor the roadway using the information that is collected and proceeds at the data centre machines.

1.3 Types of Sensor Networks

According to node capacity, technology, mobility pattern and network topology, various types of sensor networks are proposed and designed. The basic types of sensor networks are as follows:

- **Wired Sensor Networks**: in a wired sensor network, the nodes usually communicate in a peer-to-peer pattern through a wired infrastructure. Wired sensor networks are not usually resource (mainly energy and/or bandwidth) constrained because they are continuously supplied by long-period energy sources and wired communication links. Wired sensor networks are commonly used in laboratory experiment situations and/or small area/buildings where the cost of installing and cabling does not matter. However, it is not desirable to utilise the wired network in such a broad area with a large number of nodes due to the high cost of wiring, network deployment and maintenance. For example, the cost of installing and cabling of UnderFloor Air Distribution system (UFAD) (which use wired temperature sensor nodes) may be 50 to 90% of the total cost of the system [Sohraby et al., 2007]. For this reason, wired sensor network is not as popular as the wireless one for industrial and/or commercial applications.

- **Wireless Sensor Network (WSN)**: sensor nodes are networked in a WSN without any specific infrastructure or centralised control to communicate. They are usually static and communicate through provided wireless channels that are limited (in terms of communication range), unreliable and vulnerable to environmental noises, signal reflections, wireless interferences and/or physical obstructions. The main objective of WSN establishment is to provide low-cost ambient data collection services. The nodes usually are small and cheap with limited energy, computation, communication and storage resources that are able to perform only a set of basic computation and communication tasks. They sense/ measure the environmental events and then transmit the result (through single or multi-hop paths) to the consumer access point (sink) which has fewer resource limitations. WSN architecture is generally classified as either distributed or hierarchical. In the former, there is no specific distribution topology and sensor nodes are randomly scattered in the area, whereas in the latter, sensor nodes are organised in a number of separate groups such as clusters [Zhu and Zhang, 2010].

- **Mobile Wireless Sensor Network (MWSN)**: sensor nodes are attached to mobile objects to measure and report the environmental events. The network is deployed similar to WSN in an ad-hoc fashion and without any centralised control. Problematic issues and research questions in MWSN are similar to WSN because of their similarity in terms of network architecture, deployment, and wireless communication. Although resource constraint is still a problematic issue in MWSN (similar to WSN), it is not critical as much as WSN; because, firstly, more powerful resources can be attached to the mobile objects (e.g vehicles power resources), and secondly, MWSN nodes have an ability to move to re-charge area when their energy level is low. However, research in MWSN needs to give extra attention to sensor nodes' mobility patterns and the influences on the data collection and wireless communications. Vehicle sensor network is an example of MWSN in which the GPS-sensor nodes collect and report the location as well as traffic data to the sink node using GSM communications.

- **Wireless Sensor and Actor Networks (WSAN)**: a set of sensors and actors nodes are linked by wireless medium to perform distributed sensing and actuation tasks in WSAN. The sensors are usually static (rarely mobile) and have limited resources, whereas actor nodes are usually mobile and have strong energy, storage, communication and computation resources. As such the network, sensor nodes gather the environmental data/events, while actor nodes take decisions and then perform appropriate actions upon the events. Thus, WSAN allows remote and automated interaction with the environmental events. Similar to MWSN, resource constraints are not a critical issue in WSAN as the mobile actor nodes would re-charge the sensor nodes energy resources when the energy level falls under the threshold. However, providing reliable communication links between mobile actors and static sensor nodes over weak and vulnerable wireless channel is the key challenging issue of WSANs.

- **Robotic Sensor Network (RSN)**: RSN is a network of mobile robots that carry (attached) sensors around the environment to detect phenomena and produce detailed environmental assessments. It combines the key features of MWSN and WSAN and inherits the problematic issues as well as research issues from both them. RSN sensors are attached to a set of mobile nodes (robots) which are able to process the collected data, make decision on them and consequently deal with the events. However, RSN is different from MWSN and WSAN as the mobile nodes in MWSN only carry the sensors and are not able to process and make a decision on data to deal with the environmental event, whereas the WSAN's sensors are not attached to the decision maker nodes (robots) and work separately.

- **Body Area Sensor Network (BASN)**: BASN comprised of multiple sensor nodes that are attached to a human or some other creatures' body. The BASN sensors are set up near or within the body and provide monitoring, processing and communication capabilities. They measure the (human) body reactions against either internal or external events and then report the data for further processing. The collected data may provide required information that is utilised for healthcare or habitat monitoring applications. Apart from general challenges faced by the sensor network, the nodes safety, mobility patterns, compatibility and sensitivity are the challenging issues that need to be considered in this category.

1.4 Data Collection in WSN

The key objective of WSN deployment is to collect and report environmental data for data consumers [Vinyals et al., 2011]. The sensor nodes measure/sense ambient events and then transmit data to the sink which usually has sufficient resources to process and store the information. As sensor nodes usually have a limited communication range, they are not able to report data in single-hop to the sink if it resides out of reach (in terms of radio range). In this case, data packets are forwarded via a set of intermediate nodes (multi-hop paths) that relay the measured data from the event regions to the sink.

WSN Data collection is highly correlated to the network coverage and connectivity due to the sensor nodes' restrictions in communication and sensing resources. In fact, the ambient data are measured if the events are covered by the sensor nodes (located in the nodes' sensing range) and data packets are reported when the required communication links from the nodes to the sink are connected. The event sources can be scattered in the field according to two distribution models: Event-Radius (ER) and Random-Source (RS) [Krishnamachari et al., 2002]. The event occurs in a single point of the sensing field in the former (i.e 100% detection), whereas the event sources are randomly distributed in the latter (i.e random detection). Sensor data is collected by the (*source nodes*) that are around the event region and are able to sense the event source. Due to the computation and storage resource constraints and the lack of node mobility, the source nodes need to transmit/report the data to the sink for further processing and analysis through either single or multi hop wireless links. Owing to this, a set of connected wireless links need to be established from the event region to the sink to forward data packets.

WSN data reports are application dependant. This means that WSNs report ambient data according to the network applications and/or consumer requirements. There are three key models that can be used by sensor nodes to report the environmental events to the sink [Solis and Obraczka, 2006], [Yuan et al., 2003]:

1. Periodical reports: the nodes occasionally measure and report the ambient events to the sink at a set of specific intervals. Timing is one of the key factors that influences periodical reports. There are three timing models that can be used by the sensor nodes to periodically report data [Solis and Obraczka, 2004]: (1) Periodic simple: each node waits for a pre-defined period and then sends all the collected data from its neighbourhood, (2) Periodic per-hop: the node forwards the collected data as soon as it receives all data samples from its direct (single-hop) neighbours, (3) Periodic per-hop adjusted: the node adjusts its waiting time according to its location in the network. It would be longer if the node is closer to the sink and/or has a denser neighbourhood, whereas it is shorter when the node has sparser vicinity and is located far away from the sink.

2. Query-based reports: the nodes transmit the data when the sink asks to report specific events and/or particular data type. WSN queries vary according to the application, node capacity and consumer requirement. The key classification of WSN queries is [Sadagopan et al., 2005]: (1) One-shot vs. continuous queries: to report a single/discrete or continuous data flows over a period, (2) Aggregate vs. non-aggregate queries: to transmit a combined information from several nodes or a set of original values separately from each source node, (3) Complex vs. simple queries: to transmit collected data based on a single query or a set of sub-queries that are combined in an arbitrary manner, (4) Replicated vs. unique data queries: to ask either multi or a single node to respond the queries.

3. Event triggered reports: the nodes send their data when a particular event occurs in the environment.

Because of resource constraints in WSN (i.e energy, bandwidth and/or communication range), none of the reporting schemes are a very good fit. Periodical reporting periodically transmit data even if data is redundant or the consumer is not interested in it. Thus, this scheme consumes network energy for collecting and reporting data that are irrelevant or useless. The cost of query-based reporting also is high as it needs several round communications during the reporting procedure until the sink receives data. In other words, query-based report is not suitable for WSN as the intermediate nodes should transmit a number of messages to inform the sensor nodes and sink about the queries and responses. The event triggered scheme is not flexible as the data reports cannot be controlled by the consumer. The nodes repeatedly measure and transmit the sensor data as soon as an event occurs without considering the consumer interests and/or queries. This model wastes network resources as the sensor nodes report the environmental data that the sink is not interested in. By and large, it seems that none of the reporting schemes are able to collect and report data in an energy efficient manner. Owing to this, and due to the fact that the sensor nodes have restricted resources, WSN needs to utilise an appropriate technique for data collection that has the potential to reduce the transmissions/communications as much as possible.

1.4.1 Data Aggregation

Data aggregation is a technique of expressing the information in a summary form for further analysis. This technique collects data from different sources and then combines them using an aggregation function. It has the potential to decrease the number of transmissions by combining a set of data into a single packet to transmit. Decreasing the number of transmissions reduces the communication costs, bandwidth utilisation, network congestion, energy consumption and network delay in WSN. According to these advantages, and due to the node resource constraints and existing drawbacks of the reporting schemes, this technique is highly desirable to be used in WSNs. Data aggregation has three elements: data representation method, aggregation function and routing.

1. Data representation focuses on the semantic and syntactic data attributes. It should be able to provide the required details about the gathered data as summary for the consumer [Bezenchek et al., 1996].

2. Aggregation function is the method that is utilised to combine the data samples. It depends on the consumer queries and applications. Average, Maximum, Minimum, Count, Median, Rank, Standard Deviation, Variance, Sum and Regression are the key aggregation functions. The functions are classified into two main categories [Fasolo et al., 2007]: Lossy/Lossless and duplicate sensitive/insensitive. The former focuses on the possibility of recovery of the original data from the aggregated results. In other words, the function is either lossless or lossy if the consumer can or cannot, respectively, recover the original sensor data by analysing the aggregated result. Timestamp and packing aggregation are examples of lossless aggregation. In packing aggregation, the messages are packed into a packed message without any compression. On the other hand, outline aggregation, which is used in eScan [Intanagonwiwat et al., 2002], is an example of lossy aggregation function. Outline aggregation composes and aggregates a set of data like spatial energy usage and topology adjacent to provide the residual energy level of sensor nodes for the consumer. Duplicate sensitivity in data aggregation considers those functions which take account of redundant data samples in the aggregation procedure. Data samples can be redundant in a data aggregation function by chance or through valid repetition, e.g. constant temperature. Max is duplicate insensitive as involving the same value data sample for multiple times does not affect the final result. On the other hand, Average can be considered as a duplicate sensitive function as inserting redundant values for several times changes the result.

3. Data aggregation routing is the process of establishing a set of communication paths between the source nodes (which can be centralised or distributed) and the sink to forward the network (data)traffic. Routing usually includes path discovery, setting up either single or multi hop data forwarding and path maintenance. WSN routing for data aggregation should consider the following issues [Rajagopalan and Varshney, 2006]: (1) efficient resource (mainly-energy) use due

to resource constraint in WSN, (2) transmitting the maximum number of data samples to increase data aggregation accuracy, (3) minimising data collection delay by forwarding data packets through low delay paths to enhance data freshness.

There are two schemes of data aggregation routing in WSN: client/server and mobile agent [Xu and Qi, 2008], [Biswas et al., 2008]. Client/server scheme allows the source nodes to transmit their data to either the sink or intermediate aggregators to aggregate. The mobile agent model focuses on the mobile agents that move across the network to capture and combine the sensory data.

1.5 Problem Statement and Thesis Motivation

Wireless sensor networks are increasingly in use due to low deployment cost and ability to sense environmental data. The network collects environmental data using ad-hoc communications, that is without any specific infrastructure or centralised control. The sensor nodes report/transmit the sensed/measured data to the consumer access points (sink) via the wireless channels. The wireless modules only allow the nodes to communicate over limited, short radio ranges. Owing to this, routing network traffic from the event source regions to the sinks (which may not be geographically close) through single or multi-hop links is required. However, transmitting raw data samples requires the establishment of a number of communication links that consume significant amounts of network resources.

Data aggregation routing is one possible mechanism to transmit a summarised scheme of sensed data (without losing data meaning and accuracy) in a convergent fashion to the consumer access point (sink). This technique has attracted researchers' interests as it has the potential to reduce the number and size of transmissions which results in reductions to the communication overhead and consequently conserves network resources. For this reason, a number of data aggregation routing protocols ranging from sparse and small networks to large and dense ones with varying network application, topology and homogeneity are put forward in this thesis, while the protocols utilise either client/server or mobile agent routing patterns to collect and aggregate data samples.

The research in this field mainly focuses on designing/proposing data aggregation routing protocols according to a multi-objective optimisation problem that focuses on three objectives: energy saving, reducing delay and enhancing accuracy. This means that a routing protocol offers a good fit performance for WSN data aggregation if it is able to collect the maximum number of data samples with minimum energy consumption and delay. However, these objectives are defined and optimised according to the consumer requirements. For example, a protocol that is able to reduce delay and increase accuracy is well-suited to real-time applications even if it does not optimise energy efficiency. In addition, routing protocols should consider and balance the potential/existing correlations between energy, delay and the number of collected data samples (accuracy). The correlations would be introduced and discussed in Chapter 5. They influence the network performance and/or consumers' QoS (Quality of Service) requirements. For example, collecting a greater number of data samples increases energy consumption and data collection delay, especially when the data sources are randomly scattered in the network (Random-Source model [Krishnamachari et al., 2002]). Reducing delay by utilising direct communication (instead of multi-hop) increases energy consumption, whereas decreasing energy consumption using multi-hop routing will increase data collection delay [Wang et al., 2006]. For this reason, the following issues need to be addressed to design data aggregation routing protocols and enhance their performance in WSNs.

- Energy conservation: energy consumption needs to be reduced as sensor nodes usually only have limited power. Energy consumption depends on the number of transmissions, communication distance and data aggregation computation overhead. For this reason, data aggregation protocols should minimise the network traffic and path hop count.

- Reducing delay: data aggregation routing delay should be minimised as it is key for data freshness. In other words, routing delay may change – and possibly reduce – the meaning and impact of collected data on further processing at sink. Data aggregation delay depends on network congestion (network traffic), transmission distance and communication delays.

- Increasing the number of collected data items (accuracy): increasing the number of captured data samples enhances data collection robustness. Indeed, the data consumer is able to make precise decisions on the collected data if a greater number of data samples are collected. Data accuracy depends on the routing algorithm's effectiveness to report data samples to the sink.

The objective of this dissertation is to set out the design and to evaluate WSN data aggregation routing protocols in each of two models: client/server and mobile agent. It focuses on an abstraction paradigm based on the research topic as well as goals aiming to improve the performance of data aggregation routing. This dissertation considers the key performance metrics in data aggregation routing, namely energy, delay and accuracy, and then aims to balance the correlations that exist between them. This dissertation focuses on four steps that are explained in the as follows:

1. Literature review: I have carried out a systematic and exhaustive literature review on data aggregation routing in WSNs to figure out existing drawbacks in this field. First, I have studied ad-hoc routing to highlight research and challenging issues that should be considered for routing in infrastructure-less networks. Then, I have presented and explained WSN (as an application of ad-hoc network) routing protocols which support data aggregation. Finally, I have highlighted the drawbacks and research/problematic issues which need to be considered and/or investigated for WSN data aggregation routing.

2. Proposing new protocols: I have proposed two new data aggregation routing algorithms (for client/server and mobile agent model) that have the potential to tackle the existing drawbacks in this research field. To do so, I have highlighted the conventional works' drawbacks to find their causes and effects. Then, I introduced the new algorithms and predict how they resolve the existing drawbacks. The key contributions of the dissertation are listed as:(1) Data collection relevant to both event distribution models, namely Event-Radius (ER: 100% detection) and Random-Source (RS: random detection), (2) Data collection from small, medium and large networks in which the number of source nodes varies, (3) Maximising the energy efficiency in the proposed routing algorithms, (4) Reducing delay of data aggregation routing, and (5) Enhancing data aggregation accuracy by maximising the number of delivered data samples to the sink.

3. Testing and evaluation: to test and evaluate the proposed protocols, the experiments have been designed by defining specifications, criteria and scenarios such as the experiment requirements, elements and metrics, network/node characteristics, capacities and architecture, communication and computation models, data distribution schemes, data aggregation paradigms and routing core modules. Deploying a WSN with numbers of real sensor nodes would be expensive for empirical research as it needs great resources and time. For this reason, simulation is often used to test and evaluate WSN research algorithms. I have used a network simulator named OMNET++ [OMNET++, 2012] to implement and test my experiments. I have also selected four conventional works (NOID [Gavalas et al., 2010], TBID [Konstantopoulos et al., 2010], LEACH [Heinzelman et al., 2000] and Directed diffusion (DDiFF) [Intanagonwiwat et al., 2000]) as my experiment benchmarks, since they are well-known in the literature, implemented and/or simulated in real world and/or adaptable for my experimental tools.

4. Result analysis: experimental results have been analysed to figure out how the solutions could resolve the existing drawbacks in the research field. I have considered additional metrics such as network traffic and path hop count that have the potential to influence the goal parameters (energy, delay and accuracy) in data aggregation routing. In addition, I have investigated the correlations between energy, accuracy and delay. The impacts of these on data aggregation performance have been studied to evaluate the proposed algorithms and highlight what I could resolve and what should remain as future work.

1.6 Thesis Organisation

This dissertation is organised in 6 chapters as follows: Chapter 2 introduces ad-hoc and WSNs and includes the descriptions of several routing protocols in both types of the networks. It also explains the similarities and differences between ad-hoc and WSN routing to highlight the key issues which should be considered in designing WSN routing protocols. This chapter discusses routing mechanisms and techniques in both models (client/server and mobile agent) of routing which support data aggregation. Chapter 3 proposes a data aggregation routing algorithm which routes mobile agent for data aggregation in WSN called Zone-based Mobile Agent data aggregation routing (ZMA). In this chapter, the key operational phases of ZMA are justified and discussed in detail. In addition, the experimental results are presented and analysed to compare the performance of ZMA with two well-known mobile agent routing protocols specifically NOID (The Near-Optimal Itinerary Design algorithm) [Gavalas et al., 2010] and TBID (Tree-Based Itinerary Design) [Konstantopoulos et al., 2010]. In Chapter 4, a client/server data aggregation routing which called Cluster-based data aggregation routing (CBA) is proposed and discussed. The simulation results of CBA are presented, analysed and compared with two client/server protocols namely LEACH (Low-Energy Adaptive Clustering Hierarchy) [Heinzelman et al., 2000] and DDiFF (Directed DiFFusion) [Intanagonwiwat et al., 2000]. Chapter 5 focuses on investigating, discussing and analysing the potential correlations between the key performance metrics of data aggregation routing: energy, delay and accuracy. The correlations are considered to compare the performance of routing protocol in each model of routing including client/server and mobile agent. Chapter 6 summarises and concludes the research, presents its contributions and suggests the future work in this field of computer science.

Chapter 2

Literature Review

This chapter investigates wireless sensor network routing protocols which support data aggregation. It starts with a brief overview of wireless ad-hoc networks, applications and benefits, and routing protocols. Section 2.1 provides ad-hoc routing constraints, challenges, characteristics and techniques to understand and highlight routing issues in infrastructure-less networks. Section 2.2 explains routing techniques that are used by wireless ad-hoc networks and have the potential to be applied on WSNs. Section 2.3 gives a brief overview of WSN by explaining architecture, infrastructure and challenging issues it faces. In this section, various aspects of WSN are discussed with respect to routing. Section 2.4 presents general WSN routing classifications to highlight key features and techniques. Section 2.5 discusses WSN routing protocols that support data aggregation. A number of routing protocols have been proposed in which a substantial number support data aggregation. However, a small portion of them have been studied, simulated and/or implemented to be used in the real world. This section explains novelties, mechanisms and efficiency improvements to mature routing protocols which support data aggregation in either client/server or mobile agent scheme.

Routing is the process of interconnecting two or more nodes via either single or multi hop links. It includes the procedure of route discovery, establishment and maintenance. In wireless networks, the purpose of routing is to interconnect wireless source nodes which generate data to the receivers (who may manipulate the data) via direct or indirect wireless links that are logically formed through a set of intermediate nodes, relays and/or routers. Routing in wireless ad-hoc network is different from other wireless networks such as infrastructure-based wireless LAN due to four key reasons: (1) routing in wireless ad-hoc networks is performed dynamically without requiring an infrastructure. This means that there is no specific node to control network establishment in ad-hoc networks, (2) network topology may change frequently due to node/link failures and/or node mobility, (3) wireless links may have different characteristics such as delay and/or bandwidth, (4) wireless ad-hoc network nodes may have different capacities and characteristics such as constraint power and/or limited radio range. For these reasons, wireless ad-hoc network nodes need to dynamically discover, setup and maintain the routes from the source to receiver nodes. The routes should be maintained and/or updated during the network lifetime according to network topology changes that may frequently arise.

A Wireless Sensor Network (WSN) can be viewed as an application of wireless ad-hoc networks to collect ambient data for the data consumer. Routing in WSN is essential as measured data samples need to be forwarded from source regions to the sink for further processing and/or data consumer utilisation. WSN routing can be considered as a subset of ad-hoc routing due to similarities in network infrastructure deployment, communication protocols and topology change. However, it differs from ad-hoc routing due to the following reasons [Akkaya and Younis, 2005], [Al-Karaki and Kamal, 2004]: (1) sensor nodes are highly prone to failures as they are usually equipped with constrained resources especially energy, computation and communication, (2) there is no global addressing pattern (i.e IP) in WSN and so nodes usually use message broadcasting to communicate, (3) routing redundant data packets is highly avoided in WSN due to the network resource constraints, (4) WSN's dynamic nature is different from ad-hoc network as sensor nodes are usually stationary.

The key objective of WSN deployment is to collect ambient data in which sensor nodes would transmit the measured data to a single or multiple nodes, named sink, for data consumer manipulation. However, forwarding raw data packets through wireless communication links from source regions to the sink increases network resource consumption and consequently reduce the network lifetime. Data aggregation is a potential technique to reduce the number/size of transmissions and eliminate redundant data packets. Using this technique, meaningful data packets are combined to express the collected information in a summary form to the data consumer. Thus, ambient data can be collected and reported to the data consumer with low energy cost if a data aggregation technique is used during data collection. Data aggregation routing focuses on forwarding either mobile agents or data packets between source regions and sink/aggregators according to the routing model. In client/server model, routing is responsible for forwarding data packets from the source nodes to the intermediate aggregators for in-network aggregation, whereas it focuses on forwarding mobile agents to collect and aggregate data in the mobile agent model.

2.1 Ad-hoc Networks

Ad-hoc comes from a latin phrase that means "for this purpose". It is usually used for dynamic (on-the-fly) solutions that are designed to resolve specific purpose problems. An ad-hoc network is an infrastructure-less and self-configuring network that provides an arbitrary and temporary topology platform to perform computing tasks over distributed platforms [Rubinstein et al., 2006]. An ad-hoc network is deployed dynamically and without any centralised control as well as pre-existing infrastructure. Ad-hoc nodes (which are a broad set of wired or wireless electronic devices like computers, laptops, PDAs, mobile phones or sensors) establish the communication infrastructure by themselves to send, receive or relay the network traffic. The nodes may be resource constraints (i.e energy and bandwidth), so efficient network resource conservation is desirable. Ad-hoc nodes communicate to each other over either wired or wireless communication channels.

2.1.1 Ad-hoc Network Applications and Benefits

A broad range of potential applications have been proposed for ad-hoc network due to its flexibility, low-cost deployment and robustness [Hoebeke et al., 2004]. The network can be deployed in various environments and conditions specifically in the home, on skyscraper, in the desert and/or in the ocean, all without requiring any specific infrastructure and centralised control. It can also be quickly configured for emergency applications, as the network has the potential to be dynamically deployed through the provided communication channels [Ghosekar et al., 2003]. The possible applications of the ad-hoc networks can be classified into the following categories:

1. To extend existing networks: an ad-hoc network can be utilised to extend the coverage limitations of existing networks. For example, an ad-hoc network may be used to extend the coverage area of a cellular network to provide communication services. The ad-hoc nodes join the existing network nodes in a self-organising manner and then provide the network services (i.e internet access) to the area which is not covered already.

2. To interconnect networks: ad-hoc nodes can quickly deploy networks when there is no specific networked infrastructure available due to the lack of time and/or required resources. It also can be useful when the existing network infrastructure is damaged due to natural disasters like earthquakes, floods or volcanic eruptions. For example, this technology can provide interconnecting services between the rescue team, doctors and nurses who help flood victims when the local com-

munication network is damaged. The network is set up to keep the connectivity between the people who frequently move in the area and need to communicate with each other in real-time.

3. Ubiquitous computing: ad-hoc networks are suitable platforms to provide ubiquitous computing services especially in harsh conditions. The network supports communication between the embedded devices in the environment regardless of their communication capacities and physical locations. For example, ad-hoc networks can give users the ability to control their remote devices by an ipad or PDA that is connected to internet.

4. Ambient sensing: an ad-hoc network can be utilised to collect/sense environmental data. In these applications, ad-hoc nodes need to be equipped by a sensing component to collect and measure environmental data. The measured data is transmitted to the collection/aggregation point through the communication links that are provided by the ad-hoc network. The tracking of animal movements in a broad area is an example of an ambient sensing application. The nodes collect and report local data to the data consumer to enable them to monitor the animal movements or behaviours in the environment.

5. Vehicular networking: an ad-hoc network has the ability to support the interconnectivity between vehicle nodes. As the network has the ability to be deployed without any specific infrastructure, it is able to support networking amongst the nodes that repeatedly move like vehicles. For example, it can give a driver the ability to control the vehicle using an ipad that is connected by WiFi to a remote control engine.

6. Personal area networks: ad-hoc nodes can take the form of a set of portable devices which are attached to human/animal bodies to provide networking services. As an example, the network collects the data from an area and then report the result to any static access point in the network that is connected through Bluetooth.

2.1.2 Types of Ad-hoc Networks

Ad-hoc network can be categorised according to node technology and capacity, network topology and/or mobility pattern into the following classes:

- **Mobile Ad-hoc Network (MANET)** is comprised of a set of ad-hoc nodes that are mobile and communicate through wireless channels. It is an infrastructure-less network in which the nodes are fully independent to manage their own resources. The nodes communicate with each other either in single or multi-hop. Single-hop communication is provided if nodes are within the communication radio range of each other. Otherwise, the nodes can communicate in multi-hop (through intermediate nodes which relay the packets) when they are directly unreachable.

- **Vehicular Ad-Hoc Network (VANET)** is a MANET made of vehicular nodes. However, there are three differences between MANET and VANET: (1) network topology can be frequently and/or continuously changed in VANET as the vehicular nodes are repeatedly mobile and move with relatively high speed. (2) VANET nodes are usually not resource constraints because vehicular nodes are able to provide sufficient power and memory as compared to MANET nodes. (3) VANET mobility patterns are different from MANET as VANET nodes usually move through a set of streets and highways which can be defined and predictable.

- **Delay-Tolerant Networks (DTN)** is a MANET that is established in a challenged environment like space or an ocean. The difference between MANET and DTN is that DTNs (i.e spacecraft satellite communications or underwater acoustic networks) are not able to establish stable end-to-end wireless links between the nodes because of the environment characteristics. There are two techniques that is used by DTNs to enhance the stability/quality of communication in such a challenging environment [Basurra, 2012]: (1) utilising a store and forward approach at intermediate nodes that allows off-line communications between the senders and receivers, (2) message replication technique that has the potential to increase the probability of message delivery. However, the communication time/cost is increased in DTNs communications and the nodes need sufficient memory to store messages/transmissions. So, a set of parameters such as end-to-end delay, link disconnection, resource and lifetime management and/or node failure/damage are usually considered in DTN to evaluate and enhance network connectivity and quality of service.

- **Wireless Mesh Network (WMN)** is an ad-hoc network in which a mesh infrastructure comprising of mesh clients, routers and gateways is deployed. Mesh clients such as PDAs, laptops and mobile phone communicate with each other through wireless mesh infrastructure that is deployed in an ad-hoc fashion. The mesh routers are usually static and have no power restriction. They are responsible for the network backbone deployment by establishing self-configure and self-healing links in a reactive manner. Indeed, they are in charge of dealing with network topology changes caused by clients mobility and/or link failures. Gateway nodes provide the required connectivity to internet or other external wired or wireless networks.

- **Wireless Sensor Network (WSN)** is comprised of a set of sensor devices which are scattered in the field either randomly or hierarchically. They communicate to each other over an ad-hoc infrastructure to measure and report the environmental data. Similar to an ad-hoc network, there is no specific infrastructure for the sensor nodes to communicate.The network is deployed in a self-organising fashion in which each sensor independently manages its own resources. However, WSN nodes are more resource-restricted as compared to ad-hoc nodes. Unlike ad-hoc nodes which are usually equipped by powerful and sufficient resources, sensor nodes are small, cheap and have weak communication, computation and energy resources.

2.2 Routing in Ad-Hoc Networks

Routing is in charge of interconnecting ad-hoc network nodes through single or multi-hop links. Direct routing (single-hop) focuses on establishing an end-to-end wireless link between a sender and receiver node that reside in wireless communication range of each other. On the other hand, indirect communication (multi-hop) is formed by a set of logical and sequential direct paths amongst the wireless devices that may be data generator, receiver or relay nodes.

Routing is a challenging issue in ad-hoc networks as the network topology can change frequently. Ad-hoc network topology may change due to the restriction of network resources (i.e energy and/or bandwidth), the mobility of nodes, variation in velocity and the vulnerable nature of wireless communication which can be subject to fading, noise and interference [Boukerche et al., 2011]. Owing to this, communication links need to be updated or re-established to keep the network connected. For example, an ad-hoc node can re-establish a failed link or build an alternative one if the next hop node moves to out of reach area (in terms of wireless communication range) or fails due to running out of energy.

There are two components in ad-hoc networks that work in parallel to route the network traffic: the protocol mechanism and the routing matrix [Basurra, 2012]. The former defines data forwarding mechanism, data transmission scheme (i.e unicast or broadcast), routing information storage and distribution, control packets characteristics (i.e size and structure) and path discovery as well as recovery. The routing matrix works upon the protocol mechanism and its objective is path selection amongst the available ones. Route matrix finds the most optimal paths between the sender and receiver when multiple paths are available. It is in charge of making the routing decisions according to the following parameters which depend on the algorithm design or the consumer interests [Liu and Kaiser, 2005]:

1. low cost (minimum consumed energy)
2. shortest and/or minimum delay paths (fewer hop count)
3. distributed/parallel routing
4. maximising communication signal strength
5. loop avoidance
6. quick path establishment (before any further topology change)
7. maximising the throughput by minimising the packet loss ratio (maximum reliability of the links)
8. maximising network lifetime

Due to the architecture and variety of ad-hoc network applications, a great number of routing algorithms have been proposed in this field. They have been categorised into variant classes in which differences, similarities and/or superiorities are highlighted, compared and analysed [Rubinstein et al., 2006]. There are a set of features that can be considered to classify ad-hoc routing protocols: the type of routing information which needs to be acquired and maintained, routing information collection time, routing operation techniques, network architecture and/or node roles during the routing procedure.

There are two key classifications for ad-hoc routing protocols: routing protocol proactivity and operation [Basurra, 2012]. The former focuses on classifying the protocols based on the time/method of collecting routing information (i.e reactive and proactive), whereas the latter considers the operation technique that is used during routing (i.e location-awareness and multiple path establishment). The latter is perceived as a part of the former because routing protocols may utilise particular techniques based on the routing information that is reactively or proactively collected. This section considers ad-hoc routing protocols based on both classifications to highlight and compare their features.

2.2.1 Proactive, Reactive and Hybrid Classification

Proactive, reactive and hybrid is the most famous classification of ad-hoc routing. This classification is rooted in the timing features of routing information collection and maintenance. Each class of this classification is explained in the following sections:

2.2.1.1 Table-driven Routing (Proactive)

Proactive routing focuses on maintaining the route information in an up-to-date manner at nodes local routing table. The nodes maintain an updated version of the available links in their local routing tables to establish the routes proactively. Routing information can be maintained at routing tables in two schemes: link state and distance vector [Liu and Kaiser, 2005]. In the former, the nodes maintain routing information about all nodes (full topology) and then use graph-theoretic algorithms such as Dijkstra [Cormen et al., 1990] to find the requested path. In the latter, however, the nodes would consider partial topology information to establish the paths in a distributed manner. The distance vector algorithms such as Bellman-Ford algorithm [Bellman, 1956] typically have lower computational and communication overhead compared to the link state as they do not need to collect, maintain and update the global routing information of all nodes. Research in proactive routing mainly focuses on reducing route maintenance overhead, improving the adaptation of the routing algorithms to the network topology changes, and enhancing the scalability and quality of service.

Decreasing the route discovery delay is the advantage of proactive routing. A requested route is immediately established because the nodes already have the required routing information for the path in their tables. Hence, proactive routing would be particularly appropriate for time sensitive applications. However, the overhead of route maintenance is a drawback of this class. The network topology changes need to be propagated throughout the network to update the routing information which is maintained at the tables. It extremely consumes the network resources (mainly power) and need a large storage capacity to maintain routing information especially when the network is broadly deployed. The lack of scalability is another drawback of table-driven routing because the routing information needs to be proactively collected during route discovery and/or maintenance phase. Owing to this, proactive routing is more appropriate for the (wired) network in which the communication links are more stable and topology changes are fewer. The conventional works of this category focus on the adaptation, modification and the optimisation of stable network routing mechanisms for ad-hoc networks.

Optimised link state routing (OLSR) [Jacquet et al., 2001] is a link state table-driven routing algorithm that aims to decrease network traffic overhead by reducing the number of nodes that transmit the routing tables. Under OLSR, each node selects a set of its single-hop neighbours which are called multipoint relays (MPRs) to cover at-least two-hop wireless communications. The route updates are reported to MPRs that are in charge of propagating them to the network nodes using User Datagram Protocol (UDP). Compared to flooding algorithms, in which all nodes forward the routing packets, the communication overhead in OLSR greatly decreases as the number of router nodes that transmit/broadcast the routing information is reduced. However, the lack of fault tolerance is a drawback of OLSR because UDP as a communication medium offers limited error recovery services for communication errors such as link/node failures especially in MANET.

Source-tree adaptive routing (STAR) [Garcia-Luna-Aceves and Spohn, 1999] establishes a tree infrastructure (rooted in the source node) in which all required links to reach the destination(s) are provided. It selects one path (amongst all available ones) to forward data packets from source to destination based on two schemes: Optimum Routing (ORA) and Least Overhead Routing (LORA). The former selects the shortest path (minimum hop count), whereas the latter considers the route that needs minimum overhead (energy cost) to setup. Each source node discovers its own single-hop vicinity and then broadcasts all the available links as a routing table to inform its neighbours. The nodes that receive the routing table will update their own tables and then re-broadcast the routing messages. This procedure is repeated until the route request messages are received by the destination(s). As a result, a tree infrastructure is established from the source to the destination nodes in which all possible paths are provided. The destination would reply the source node through a minimum hop count or energy cost path according to the route selection criteria (ORA or LORA).

Destination-Sequenced Distance-Vector (DSDV) [Perkins and Bhagwat, 1994] is an extended version of original Bellman-Ford algorithm [Bellman, 1956] that proactively routes network traffic according to the stored information at the nodes. Each node maintains a routing table containing (discovered) destination's ids, their distance (hop count) and a sequence number of route request message to find fresh routes and avoid loops. Routing tables can be updated in either time-driven or event-driven scheme. The former broadcasts route update messages at a particular time, whereas the latter forwards them if a particular event (i.e topology change) arises. Route update messages are classified into two categories: full dump and incremental. Full dump messages transmit the full routing table that contains all collected routing entries, whereas incremental sends just changed and/or updated entities.

Table 2.1 compares and summarises the key features of OLSR, STAR and DSDV.

Table 2.1: Key features of proactive routing protocols

Features	OLSR	STAR	DSDV
Mechanism	Link state	Link state	Distance Vector
Architecture	Flat	Hierarchical (tree)	Flat
Route Updates	Periodical Hop-by-Hop	Event-based Hop-by-Hop	Periodical/Event-based Hop-by-Hop
Communication Type	limited broadcast through MPRs	limited broadcast over tree infrastructure	full broadcast
Key advantage	reduced number of route broadcasters	reduced overhead of communications (over tree)	fresh route utilisation
Key drawback	limited error recovery	high cost of infrastructure establishment and maintenance	high cost for full broadcast route updates
Routing metric	shortest path	shortest path	shortest path
Loop-free	yes	yes	yes

2.2.1.2 Source-initiated Routing (Reactive)

Reactive routing establishes a route when a source node needs to communicate with a particular destination node. The source node broadcasts the route request and waits then until the reply packet is received from the destination. After route discovery, reactive routing needs to perform a route maintenance phase to update the existing paths based on the network topology changes. Reactive routing offers two benefits compared to proactive: (1) routing overhead is reduced due to two reasons: first, the size of updating area is limited to specific regions in which the routes are requested instead of whole the network. Second, as reactive links are established on-demand for temporary usage, the source nodes do not need to maintain a path after transmitting data packets. It results in reductions to path maintenance cost. (2) higher scalability is achieved because reactive routing has the ability to adapt itself to network changes by establishing on-demand routes. Routing delay is a drawback of reactive routing. Source-initiated routing protocols increases the routing delay as the paths are dynamically selected on-demand. Reactive routing delay varies depending on the availability of nodes and communication links on the route, the number of intermediate nodes, the speed/frequency of topology changes in the network and the network architecture and application. Reactive routing research usually focuses on optimising link failure recoveries in terms of speed and efficiency to select stable routes, reducing routing cost and improving routing performance by considering additional metrics such as quality of service and/or signal strength.

A global network addressing scheme is required to support peer-to-peer communications between the source and the destination nodes in reactive routing. Global addressing schemes (IP-based networks) give the source nodes the ability to find the requested destination nodes to establish the paths. However, utilising this technique is not feasible in large-scale ad-hoc networks that consist of weak nodes (i.e WSN) as the cost of maintaining and updating the global address tables is extremely expensive.

Dynamic source routing (DSR) [Johnson et al., 2001] forwards routing messages from the source nodes to the destinations to establish paths in a broadcast manner. It has two phases: route discovery and maintenance. In the first phase, routing messages are broadcasted by the source nodes to the requested destination to discover available routes. The messages are updated at intermediate nodes until they are received by the destination. Each node that receives the route request firstly checks its own routing table to find the requested path to the destination. If the path is not found, the node adds its address to the table and forwards the route request until it is received by the destination. The destination sends back a route reply packet to the source node through the same path when the route request is reached. Finally, the source node forwards the data packet through the established path when it receives the route reply packet from the destination. Source nodes maintain multiple routes to the destination node as alternatives. At the maintenance phase, DSR ensures the validity of existing paths using route error and acknowledgement messages. If any node detects a link failure, it sends a route error packet back to the source node to inform it that the current link is going to fail. The source node establishes another path or use alternative ones to forward data packets through to the destination. Stable Weight based On-demand Routing Protocol (SWORP) [Wang et al., 2007a] improves the performance of DSR by allocating three metrics, namely expiration time, error count and hop count, to the potential links between the source and destination nodes as link weights. Hence, the destination node selects the route which is optimal (has the biggest weight) amongst all the established links.

Ad-hoc On-demand Distance Vector (AODV) [Perkins and Royer, 1999] forwards routing messages from the source nodes to the destinations in a unicast or multicast manner to establish paths. The routing messages maintain the information of active links instead of all available ones. It reduces the routing cost to a great extent. The messages contain: the address of the source node, the current sequence number of the route request message, the address of destination node and the sequence number of the last route request message (for the same destination). The intermediate nodes (receiving route request messages) reply to the source node if any record for the requested path is found at their local routing tables. Otherwise, they keep a version of the route request at their local tables and then forward the route discovery message for the next hop nodes. It is repeated until the destination is reached. Similar to DSR, the source node transmits the data packet through the established path when the route reply packet is received from the destination node. The difference of DSR and AODV is that the latter locally maintains the routing information of the forwarding path in each node routing packet, whereas the former puts them all discovered paths in the route discovery messages. Owing to this, the traffic overhead is reduced in AODV due to transmitting smaller size control/routing packets [N.Thakare and Joshi, 2010]. Moreover, routing overhead of AODV would decrease, in comparison to DSR, as AODV does not need to send the information of all available routes to the destination. However, AODV routing is less fault-tolerant as it maintains a single route instead of multiple ones from source nodes to the destinations.

Signal stability-based adaptive routing (SSBR) [Dube et al., 1997] considers two metrics namely signal strength and location stability during routing to enhance reliability. SSBR has two components: Dynamic Routing Protocol (DRP) and Static Routing Protocol (SRP). DRP utilises hello messages to measure signal strength. Received Signals Strength (RSS) are maintained at signal stability table of each node. The tables can be frequently updated when the nodes receive any further packets from the neighbour nodes in the case of network topology changes. SRP is responsible for performing routing. It utilises signal stability tables to forward data packets through the links that are more reliable and have stronger RRS value. SSBR decreases the routing overhead as compared to AODV and/or DSR. It selects routes based on the link strength instead of hop count that are used in AODV and DSR. As a result, the control overhead of routing is reduced as it does not require extra control packets to consider hop count. However, SSBR does not support local/partial route discovery. In fact, SSBR should re-discover requested routes from the source to the destination node if any intermediate node fails. Moreover, SSBR is not useful to work over long period as wireless channel strength may vary according to the nodes location, environmental effects (i.e noise) and the level of residual power.

Temporary Ordered Routing Algorithm (TORA) [Bakht, 2011] establishes shortest paths to forward data packets by considering distance value between source and destination. First, it maps the network into a directed graph, which maintains all possible routes from the source nodes to the destination. The graph entries are assigned by a metric called height that is the link distance value. In fact, it is a proportional level/value of the node proximity to destination. Farther nodes have greater height values, whereas closer ones have lower height. To get the height value, a source node sends a path query message containing its own and the destination id. The route query message is forwarded by intermediate nodes until it is received by either the destination or any of its single-hop neighbours. Upon receiving the route request, the destination sends back an updating message containing node height to the source node. The value increases at each intermediate node and the reply message is forwarded until it is received by the source. At the end, a path is created from the source node

Table 2.2: Key features of reactive routing protocols

Features	**DSR**	**AODV**	**SSBR**	**TORA**
Mechanism	Hop-count	Hop-count/route sequence	RSS	Hop-count
Architecture	Flat	Flat	Flat	Flat
Route Updates	as needed	as needed	week/lost signals	link/node failures
Communication Type	broadcast	uni/multicast	broadcast	local broadcast
Key advantage	1- Multiple routes 2- low complexity of implementation	reduced routing overhead and/or table size	reduced control overhead	reduced overhead for route updates
Key drawback	1- High overhead 2- high bandwidth utilisation	single route maintenance	not adaptable with varying signals over time	high resource consumption in large/dense networks
Routing metric	shortest path	freshest and/or shortest path	reliable path	shortest path
Multiple Routes	yes	No	yes	yes

to the destination via intermediate nodes whose height values are decrement. The benefit of TORA is its ability to deal with network topology changes. Any node that detects a failed link removes the path by setting its own height to a reference rate (highest height) and then broadcasts a message to inform all its neighbours about the failure. The record of a failed link is discarded at each node that receives the failure update message. However, memory usage and communication overhead is increased in TORA as compared to AODV especially in a dense and large network in which network topology is frequently changed.

Table 2.2 compares and summarises the introduced reactive routing protools.

2.2.1.3 Hybrid Routing

Hybrid routing combines the key aspects of source-initiated and table-driven routing protocols to enhance routing performance. The idea of combining proactive and reactive routing is that table-driven routing algorithms are suitable for the network with more stable connections, whereas source initiated protocols are appropriate for the network with high topology changes [Liu and Kaiser, 2005]. Hybrid routing inherits the existing drawbacks of both source-initiated and table-driven routing that are high delay and overhead respectively. Reactive route establishments increases delay, whereas proactive paths (that need to be updated according to network topology changes) increases routing overhead [Boukerche et al., 2011].

In a nutshell, there are two techniques that usually are used by hybrid routing:

1. Proactive zones: the network is partitioned into a set of zones in which intra-zone paths are formed in proactive, whereas inter-zone packets are reactively routed. The objective of this technique is to limit the proactive routing into restricted zones to reduce routing overhead. In other words, the overhead of routing is reduced as a fewer number of communication links (intra-zone paths)

are proactively discovered, established and/or updated. The zones can be formed according to link stabilities and/or node similarities (i.e mobility pattern and physical location). For example, Landmark Ad-hoc routing (LANMAR) [Pei et al., 2000] forms a set of proactive zones based on the nodes similarity in mobility patterns such as velocity and direction. However, there is a trade-off between energy consumption and routing delay in forming proactive zones. Although forming broad zones reduces delay to some extent (a greater number of nodes use proactive routing), it increases communication overhead/cost to proactively establish and update the intra-zone links. Owing to this, zone count, size and density need to be considered to balance the trade-off.

2. Network backbone: the nodes that have more stable connectivities form the network backbone (i.e tree) in which the packets are proactively forwarded, whereas other ones route the packets reactively. The objective of this technique is to limit reactive routing to the area which is not covered by the network backbone. The packets are forwarded reactively until they are received by backbone nodes that provide proactive routing. It reduces communication routing delay as the backbone would be able to quickly forward packets from/to different parts of the network. However, the network backbone needs to balance the delay-energy trade-off similar to the proactive zones. The backbone should cover the broadest area of the network to limit reactive routing and consequently reduce routing delay. On the other hand, the cost of establishing and updating communication links on the network backbone needs to be minimised by reducing the number of node/links that reside on the backbone. As a result, the researchers need to design hybrid protocols in which the backbone nodes cover frequent use parts of the network with a minimum number of nodes/links.

Zone routing protocol (ZRP) [Samar et al., 2004] utilises proactive zone technique in which network packets are routed using two algorithms: Intra-zone Routing Protocol (IARP) and Inter-zone Routing Protocol (IERP). The former is responsible for intra-zone proactive routing, whereas the latter focuses on reactive inter-zone routing. ZRP firstly forms the network zones around each node according to a hop count value which is called allowed hop number. The allowed hop number defines zone size and is set by the network consumer. The nodes perform IARP to discover their local neighbourhood (with respect to allowed hop number) by exchanging hello messages. Then, Inter-zone Routing protocol (IERP) is performed by only the nodes that reside on the border of zones. The packet is routed from the source to the destination if both reside on the same zone. Otherwise, the route query message is forwarded from the source to any other zone that knows the destination. The nodes that reside on the border of zones (broader nodes) establish reactive links to transmit the route messages between the zones. ZRP reduces the overhead of reactive routing as a limited number of nodes (zone boarders) instead of all nodes that receive the messages are in charge of establishing reactive routes. However, zone size (or allowed hop count) is the key drawback of ZRP. Network resources is extremely consumed to collect intra-zone information if zone size is set too large. On the other hand, delay and/or the overhead of IERP is increased when the zones are formed small in which a great number of nodes reactively establish the inter-zone routes.

Landmark ad-hoc routing (LANMAR) [Pei et al., 2000] forms the proactive zones to forwarded data packets according to nodes mobility patterns. This means that LANMAR logically partitions the network into a set of subnets in which the nodes have same mobility patterns like direction and/or velocity. In each zone, a node (called landmark) is selected to manage connection status of links and node locations. Each node is allocated by a routing table in which the zone number, landmark Id, and the list of neighbours are maintained. Routing tables are updated using route update messages that are propagated in the case of network topology changes. An intermediate node replies back to a route request message if it can find any record of destination in its own routing table. Otherwise, it re-sends the query to its landmark node to find and establish a route to the destination. Landmark nodes need to communicate with each other to find the destination. The landmark availability is a drawback of LANMAR as landmark nodes should be reachable for all nodes in their zones to monitor local topology changes and process remote route requests. Moreover, routing performance and accuracy is threatened when connections between nodes and their landmarks fail. Thus, landmark nodes would not be able to provide required information for the source nodes to find the destination. Besides, network resource consumption and delay is significantly increased when the network is highly mobile as the landmark nodes should be frequently updated according to network topology changes.

Zone based hierarchical link state routing protocol (ZHLS) [Joa-Ng and Lu, 1999] forwards packets over a zone-based network in which the zones are formed according to the physical location of nodes. The network is partitioned into the zones using a zone id which is allocated to each node via GPS. Then, each node collects routing information proactively from its neighbours that have the same zone id. The routing information is used to forward intra-zone messages when source and destination nodes reside in same zone. Otherwise, the source node forwards a zone level query to find any node of the destination zone. Data packets are forwarded by the source when a route reply packet is received from either destination or any node in the destination

Table 2.3: Key features of hybrid routing protocols

Features	**ZRP**	**LANMAR**	**ZHLS**
Mechanism	Hop count zone	mobility pattern	GPS zone
Architecture	Flat	Hierarchical	Flat
Route Updates	Periodical	Event-based or node mobility	Periodical
Communication Type	Local/limited broadcast (zone boarders uni/multicast)	limited broadcast (Landmark uni/multicast)	limited broadcast (GPS communications)
Key advantage	1- reduced message forwarding 2- flexible	mobility supported	quick routing
Key drawback	optimal value for allowed hop number	landmark availability, establishment and maintenance	GPS communication is expensive for MANET
Routing metric	shortest path	shortest/freshest path	shortest path
Loop-free	yes	yes	yes

zone. ZHLS is expensive for a MANET network as the nodes are highly mobile and should be frequently allocated by new zone IDs. It increases network traffic. Moreover, intra-zone routing tables need to be continuously updated due to the topology changes caused by mobile nodes. Each inter-cluster movement needs to be recorded at the routing tables of zone members. In addition, this protocol is not suitable for an ad-hoc network in which the nodes are resource constraints as the communication overhead with GPS is usually expensive. The zone id generator (GPS) should be continuously reachable for all the nodes that may have limited communication radio range.

The introduced hybrid routing protocols are compared and summarised in Table 2.3.

2.2.2 Routing Operation Classification

Ad-hoc routing protocols can be categorised based on routing mechanisms that are used to discover, establish and/or maintain communication links. The mechanisms are used to enhance routing performance by resolving existing drawbacks and/or reducing routing operation limitations. However, the drawbacks of proactive/reactive classification (delay of reactive and cost of proactive routing) is inherited depending on the time/method that is used to collect routing information. This section focuses on explaining three key categories of routing operation classification namely location-aware, multi-path and power-aware.

2.2.2.1 Location-aware

Location-aware routing utilises the nodes location information to route the packets. The performance of routing can be enhanced as the location information gives the source nodes a clear vision about the potential paths to the destination(s) in the network. The physical location information can be collected by the nodes from their local vicinity (i.e clusters), the whole network (networked regions) or external location information providers (GPS). The information can be gathered reactively or proactively based on the application and node characteristics. For example, static nodes can proactively collect the location information, whereas mobile nodes need to reactively collect the location information on-demand. The research in this class of routing mainly focuses on reducing routing delay and/or improving routing accuracy using the location information.

The key advantage of location aware routing is that the routing delay is decreased. As the nodes would be able to detect the destination nodes using location information, the routes are established quickly and without additional delays. However, the overhead of collecting the location information can reduce the quality of routing performance especially in mobile network. The nodes should frequently gather location information to update routing tables according to the network topology changes.

The easiest method to collect the geographical information of nodes is the use of GPS [Boukerche et al., 2011]. However, communicating with GPS to get the information is not suitable for a network which has limited communication and energy resources (i.e WSN). In this case, the nodes should consume a great deal of network resources to get the information from GPS especially during the updating phase which caused by topology changes. Besides, the availability of GPS is still a problematic issue. The GPS should be reachable for all network nodes and the nodes should have sufficient power capability to communicate with location information provider.

Location-aided routing (LAR) [Ko and Vaidya, 2000] is a reactive location-aware routing protocol that aims to reduce routing overhead by minimising the route discovery area. The sender node uses GPS to collect location information (location, movement direction and speed) of the receiver. LAR essentially utilises two schemes to forward data packets: first, it detects the exact region that is centred by the destination node in the network using the collected location information. The route request packets are forwarded to destination through any node that resides in or connected to the destination region. Second, the source node calculates its distance to the destination using the information provided by GPS. Then, the node broadcasts the route request packet that is assigned by the distance value. The packet is forwarded by any node that has a shorter distance to the destination until it is reached by the requested destination. LAR reduces routing overhead as a limited region of network is considered to forward data packets instead of the whole network.

Distance Routing Effect Algorithm for Mobility (DREAM) [Bakht, 2011] is a proactive table-driven routing protocol in which distance information of all nodes are collected and maintained at each node. Using this information, the nodes would be able

Table 2.4: Key features of location-aware routing protocols

Features	**LAR**	**DREAM**
Mechanism	Geographical location/distance	Distance
Architecture	Flat	Flat
Proactivity	Reactive	Proactive
Communication Type	Local/limited broadcast	broadcast
Key advantage	quick as routing search is reduced	mobility is supported
Key drawback	GPS availability	expensive route update
Routing metric	freshest/shortest path	shortest path
Loop-free	yes	yes

to find the location/region of the destination node to immediately forward the route request packets. The source node sends data packets to the nodes that reside closer to the destination. This procedure is repeated until the destination is reached. The routing table at each node is updated according to network topology changes such as node movement, node failure and link expiration. Although the routing search domain is reduced in DREAM similar to LAR, network resource consumption is increased when the network deployed is dense and large. The nodes need large storage capacities to maintain the routing information (distance) of the network nodes and also to consume a great deal of network resources (i.e bandwidth and energy) to handle route update messages that are frequently transmitted.

Table 2.4 compares and summarises the key features of LAR and DREAM.

2.2.2.2 Multi-path

Multi-path routing provides multiple routes to enhance routing reliability. In comparison to other routing categories that usually establish a single path from the source to destination, the ratio of message delivery is increased in multi-path routing as establishing multiple paths enhances the chance of path availability. Indeed, source nodes would be able to transmit data through any other available path when the current one is not available (due to collisions and/or link failures). Multi-path routing can also decrease the data transmission delay by distributing the traffic overhead on multiple paths that interconnect the source and destination node pairs. This means that, data packets can be transmitted through any other available route to the destination if the current route is busy or fails. The drawback of multi-path routing is the additional overhead that is required to establish/maintain the multiple routes. This overhead is increased when the network nodes are highly mobile as a number of paths need to be re-established according to the network topology changes. Network loops are another drawback of multi-path routing. Loop probability is increased due to establishing multiple paths that can be partially interconnected. This drawback can be resolved by considering link sequences (list of nodes residing on the link) at intermediate nodes. However, it increases routing overhead when the network is deployed large and/or dense. Multiple routes maintenance can also increase network resource consumption (i.e energy and bandwidth) as the availability of paths need to be frequently considered. As a result, multi-path routing is unsuitable for an ad-hoc network in which the nodes are highly constrained by resources (i.e WSN).

Multi-path Security-aware QoS Routing (MuSeQoR) [Sriram et al., 2005] considers Quality of Service (QoS) metrics (i.e signal strength) to establish multiple routes between source and destination nodes. The source node broadcasts a route query containing a sequence number, the list of visited nodes, path reliability value (QoS metric) and destination ID. Each intermediate node (receiving the route query packet) adds its id as well as the quality/reliability of the last established link (based on the QoS metrics such as the strength of the received signal) to the packet. The route request packet is re-broadcasted until the destination is reached. The destination receives the route query packet through different routes. Thus, it selects the most reliable path by considering the reliability value that is assigned to each receiving route request. Other route requests (ranked abased on the QoS metric reliability) are maintained to be utilised if the current route fails and/or is busy. MuSeQoR can be considered as a fault tolerance routing protocol for network with high mobility and/or vulnerable links. It is also loop-free as the intermediate nodes would be able to detect the loop using the route request sequence number. However, MuSeQoR increases network resource consumption (i.e bandwidth, memory and energy) in dense and large networks as the size of route requests is increased to keep the required information of each discovered path.

Split Multi-path routing (SMR) [Lee and Gerla, 2001] discovers multiple node-disjoint paths from the source to the destination nodes in which the route with minimum delay is selected to forward data packets. First, the source node floods a route request packet into the network (re-transmitted by the intermediate nodes) until the destination receives. The destination node maintains a record of all the established paths and then replies back the source node through the path which has minimum delay. The main difference between SMR, AODV [Perkins and Royer, 1999] and DSR [Johnson et al., 2001] is that destination is the only node in SMR that is able to reply back the route request. In SMR, the intermediate nodes cannot reply back to the source node individually even if they have a reserved path to the destination. SMR is unsuitable for dense and large ad-hoc network (i.e WSN) as it works based on message flooding which increases network resource consumption.

Table 2.5: Key features of multi-path routing protocols

Features	MuSeQoR	SMR
Mechanism	Quality of Service paths	Minimum delay links
Architecture	Flat	Flat
Proactivity	Proactive	Reactive
Communication Type	broadcast	broadcast
Key advantage	fault tolerance is increased	simple to implement
Key drawback	increased size of routing messages	expensive route discovery/update
Routing metric	most reliable path	low delay path
Loop-free	yes	No

Table 2.5 compares and summarises the key features of discussed multi-path routing protools.

2.2.2.3 Power-aware

Power-aware routing forwards the network packets according to the available level of energy at nodes. Power awareness needs to be considered as a key issue in wireless routing because the quality of communication depends in large part on the residual power at nodes. This means that the link would be reliable to transmit data when the nodes have enough energy to send or receive high quality wireless signals. The objective of power-aware routing is to forward data packets through the nodes that have enough energy to provide required level of signal strength for communications. As a result, power-aware routing needs to consider the following parameters during routing procedure [Boukerche et al., 2011]: (1) decreasing the traffic overhead and communication rate to reduce the message cost, (2) reducing the computation loads of routing by removing unnecessary and redundant tasks, (3) increasing network lifetime by distributing the routing workloads on the network nodes uniformly instead of focusing on a set of specific nodes in the network (i.e cluster heads).

This class of routing is suitable for a wireless network that needs reliable communications. It enhances message delivery as the nodes select the neighbours that have sufficient energy and are able to cast strong wireless signals. In other words, power-aware routing would aim to reduce the ratio of node/link failures and transmit network packets via routes that have a higher reliability. However, power-aware protocols are not suitable for a network that needs to consider Quality of Service (QoS) parameters such as delay. For example, routing delay is increased when the nodes transmit the data through reliable and/or low-cost paths that are not necessarily short in terms of hop count.

Online energy aware routing [Mohanoor et al., 2009] is a proactive algorithm

Table 2.6: Key features of power-aware routing protocols

Features	**Online energy aware routing**	**CLUSTERPOW**	**MINPOW**
Mechanism	Dijkstra	Clustering based on energy	Bellman-Ford
Architecture	Flat	Hierarchical	Flat
Proactivity	Proactive	Hybrid	Proactive
Communication Type	uni/multicast	limited broadcast	broadcast
Key advantage	minimum energy consumption is reduced	reliable inter/intra-cluster communication	reduced energy over multi-hop links
Key drawback	reduced routing accuracy due to varying energy level over time	bottleneck at CHs	reduced routing accuracy due to varying energy level over time
Routing metric	minimum consumed energy	shortest/freshest path	maximum residual energy

which considers three parameters to establish the routes: residual energy, the minimum required energy and total consumed energy level. In this algorithm, the network is mapped into a weighted graph whose edges are marked by a weight value that shows the required power to transmit a network packet through. Dijkstra algorithm [Cormen et al., 1990] is utilised then to find the shortest (minimum consumed energy) path on the graph. It selects an energy efficient path between the source and destination node by calculating and minimising consumed energy which is required to transmit network packets. The drawback of this protocol is that collecting energy information for routing in a proactive manner reduces routing accuracy and performance as nodes residual energy level would change over time.

CLUSTERPOW and MINPOW protocols proposed by Kawadia and Kumar [Kawadia and Kumar, 2003] establish paths that have maximum transmit power to forward network packets. CLUSTERPOW proposes a clustering algorithm in which the Cluster-Heads (CHs) are selected based on residual energy level (instead of traditional clustering parameters such as coverage degree and/or physical locations). CHs are responsible for collecting intra-cluster route requests and performing inter-cluster routing. A CH replies to a route request if it knows the destination. Otherwise, the CH forwards the route request to the cluster-head residing in the same cluster with the destination. MINPOW does not establish a hierarchical infrastructure to route network packets. Similar to DSDV [Perkins and Bhagwat, 1994], MINPOW establishes the paths using a modified Bellman-Ford algorithm [Bellman, 1956] in which total consumed energy over a multi-hop path is considered instead of hop count. It selects the path in which the residual energy level is maximised.

The key features of introduced power-aware routing protools are compared and highlighted in Table 2.6.

2.2.3 Summary: Ad-hoc Network Routing

Routing in ad-hoc network is a challenging issues as it can be frequently influenced by network dynamism and/or topology changes. Ad-hoc networks are usually deployed in a self-organising manner and without any centralised control. The network topology can be changed due to node mobility, resource limitation, variation in velocity and vulnerable nature of wireless communication. Hence, the route connectivities would depend on the availability of links and/or nodes that are being used. In other words, a route is connected to the nodes/links that are alive/available to transmit the network packets. Due to these issues, ad-hoc routing aims to reach a set of key objectives that are summarised below:

1. The route stability needs to be enhanced by establishing/re-establishing communication links as quick as possible before any further topology change.
2. The reliability of routes/links need to be enhanced by transmitting network packets through nodes that have sufficient residual energy and/or signal strength to communicate.
3. Network lifetime needs to be maximised by minimising energy consumption. It can reduce node/link failures that are caused by running out of energy.
4. Network resource consumption needs to be minimised as ad-hoc nodes may be equipped by limited resources.
5. Routing workloads need to be processed in a distributed/parallel manner to uniformly consume network resources. This reduces the probability of partial node failures caused by bottlenecks.

A number of routing protocols are proposed in this field to improve routing stability, scalability and extendibility. They are categorised into variant classes based on features, techniques and objectives that are considered/utilised during routing procedure. Table 2.7 compares and summarises the over-riding ad-hoc routing categories.

2.3 Wireless Sensor Networks

Wireless sensor networks (WSN) are considered an application of ad-hoc networks in which nodes are able to collect/measure environmental data. Similar to ad-hoc networks, there is no specific infrastructure for WSN and the network is deployed in a self-organising manner without any centralised control. However, there are five differences between WSN and ad-hoc networks [Krishnamachari et al., 2002]:

Table 2.7: Comparison summary of ad-hoc routing categories

Category	Objective	Mechanism	Pors	Cons
Proactive	quick routing	up-to-date routing table	reducing routing delay	1- lack of scalability 2- not flexible to topology change 3- high table maintenance/update overhead
Reactive	resource conservation	on-demand routing	improving scalability	increased delay
Hybrid	balancing energy-delay trade-off	1- proactive zone 2- network backbone	similar to reactive/proactive	similar to reactive/proactive
Location-aware	1- reducing routing overhead 2- partial/local routing	geographical/physical location information	1- reducing delay 2- increasing routing accuracy	1- overhead of collecting location information 2-availability of location info. provider
Multi-path	enhancing routing reliability/availability	multiple links	1- increasing message delivery 2- reducing collisions	1- extra overhead for multiple links 2- increasing the probability of loops
Power-aware	enhancing reliability	considering residual level of energy	1- energy conservation 2- reduced link failures	QoS may be avoided

1. WSNs are densely deployed using a large number of sensor nodes, whereas ad-hoc networks usually consist of a fewer number of nodes with sufficient resources to compute and/or communicate.

2. Sensor nodes are usually equipped with weaker resources (energy, communication and computation) as compared to ad-hoc nodes. In other words, sensor nodes are tiny, weak and cheap (i.e TelosB node [Williams, 2014]), whereas ad-hoc nodes are usually more mature (i.e smart phones and/ PDAs) and have stronger resources. Sensor nodes are not able to perform complex algorithms and process continuos/high-rate data streams as they are usually equipped by restricted processing modules (CPU). They are not able to maintain large-scale data because they have limited storage capacities. In addition, they are not able to frequently communicate over long distance wireless links as they have limited radio range and power resources to broadcast wireless signals.

3. Message broadcasting is usually used in WSNs as maintaining a global addressing pattern such as IP (to support peer-to-peer communications) is very expensive in terms of network resource consumption. On the other hand, an ad-hoc network is able to support local communications between any pair of nodes using IP-based communication protocols.

4. In contrast to ad-hoc, WSN avoids collecting and transmitting redundant data as it increases network resource consumption.

5. WSN nature of dynamicity is different from ad-hoc network as sensor nodes are usually stationary in most applications.

A WSN is usually deployed by a large number of sensor nodes that are scattered to collect and report ambient data. There are four components that need to be considered to deploy a WSN [Sohraby et al., 2007]: (1) a set of localised or distributed sensor nodes that are responsible for sensing/measuring ambient data and reporting/transmitting the results to the network consumer access point. In most WSN applications, the nodes are static. Deploying mobile nodes is usually expensive for the engineers/researchers as WSNS are usually deployed by a large number of disposable nodes that are scattered in unattended and/or inaccessible area(s). (2) a networked infrastructure that interconnects the nodes. The network is usually deployed in an ad-hoc fashion and without any centralised control to transmit measured data samples from the sensing area to the network consumer access points. (3) centralised and/or distributed data sources. They are basically a set of event sources that should be monitored by the sensor nodes. They generate (ambient) data for the sensor nodes to measure. They can be mobile (i.e tracking object) or static (i.e temperature). (4) centralised and/or distributed computing points that are used to capture, process and analyse reported data. The nodes usually have powerful computation/storage resources that allow to store and/or process large streams of raw data. They may also have sufficient communication resources to deliver the collected results to network consumers either directly or using external telecommunication infrastructures such as the internet and/or satellite. These nodes are called sinks.

WSN has attracted researchers attention because it allows them to measure, collect and report ambient data in an easy/cheap manner without requiring any specific infrastructure, centralised control and administration. The unique features of WSN let the researchers design a broad range of applications such as field surveillance and disaster management. It allows them to collect and report ambient data from harsh and hostile environments that are not easily and/or fully reachable. WSNs are designed according to the network consumer applications and/or requirements. The features of WSN design and deployment are listed below.

- Small node size: reducing node size has the potential to facilitate network deployment and resource conservation as WSNs are usually deployed unattended in harsh or hostile environment with large number of nodes.

- Low cost nodes: unattended sensor network deployment in hostile environment means the nodes cannot be reused. Owing to this, reducing node cost would be essential as it consequently reduces the network deployment cost.

- Low power consumption: re-charging sensor nodes power sources (batteries) is usually difficult/impossible as they may be scattered in out of reach or harsh environment. Therefore, it would be crucial to reduce power consumption in a WSN to enhance the network lifetime.

- Scalability: sensor networks should be scalable as they can be deployed in different sizes of tens, hundreds or thousands. The sensor nodes also may be added to the network which is already deployed.

- Reliability: sensor network protocols need to provide error control mechanisms to ensure reliable data delivery over noisy, error-prone and/or time-varying wireless channels that are used to transmit data in unpredictable environments.

- Self-configurability: WSNs need to have this ability as they are deployed unattended and without external administrations. Network deployment in hostile and unpredictable environments, and nodes resource constraints lead WSNs to be extremely prone to failures. It would be difficult (for the administrators) to re-configure the network resources once the network is deployed as it may be inaccessible. Owing to this, a WSN should be able to automatically organise itself in the case of topology changes and/or node failures.

- Adaptability: it would be required that a WSN has the ability to adapt itself according to the changes caused by frequent/unpredictable network topology, environmental conditions, and/or network size/density.

- Channel utilisation: WSN communication protocols should efficiently use the network bandwidth and enhance channel utilisation as the network is usually provided with limited bandwidth resources.

- Fault tolerance: sensor networks need to be fault tolerant (having self-testing, self-calibrating, self-repairing and self-recovering abilities) as they are highly prone to failures especially when deployed unattended in harsh environments.

- Security: security is required in such the networks to prevent network information from unauthorised access and malicious attacks as the network is usually deployed in unpredictable and/or unsafe environments.

- QoS support: WSN applications (or consumers) may specify a set of QoS requirements (i.e latency and/or packet loss ratio) according to the consumer interests.

2.4 WSN Routing

In a nutshell, routing in WSN focuses on forwarding data packets from source nodes to the data consumer access point (sink) according to a convergence pattern through either single or multi-hop links. Sensor nodes may need to forward network traffic through multi-hop links (intermediate nodes) to sink(s) as they usually have limited communication abilities which do not allow direct communications. As a result, WSN routing needs to efficiently route the data packets from source regions to sink based on the network characteristics, node capacities and application requirements.

The paths are established in WSNs in two schemes: Address-Centric (AC) and Data-Centric (DC). In the former, the nodes consider the address of next hop nodes to forward network traffic, whereas in the latter the routes are established using an attribute-based naming that specifies the properties of data over the wireless links. Address centric protocols are not advantageous in WSNs due to the lack of global addressing scheme (i.e IP). In fact, lack of global addressing scheme in WSNs limits address-centric communications into local area in which sensor nodes are aware of each other ID address. Moreover, dense, dynamic and/or random WSN deployment complicates acquiring ID address of nodes on multi-hop communication links. As a result, WSNs would need to use data-centric routing protocols to forward data packets from data regions to sink. In data-centric routing, a data packet is forwarded if it is desirable for the next node.

Network topology change is an issue that have high impact on WSN routing. In WSN, node availability, wireless signal quality and efficient wireless communication range depend on the nodes residual energy level. A node or communication link/route fails when the energy level at the nodes falls below the required threshold for being alive or maintaing the communication links connected. For this reason, routing in WSN is a challenging issue in which the sensor nodes need to consider network resource consumption (mainly energy), connectivity and coverage to route network traffic from (data) source regions to the sink(s) [Srivastava, 2010].

2.4.1 Comparison of Ad-hoc and WSN Routing

A wireless sensor network (WSN) is an infrastructure-less and self-configuring network (similar to ad-hoc networks) comprising of a number nodes that are distributed in the field to collect and report ambient data. There is no centralised control and pre-existent infrastructure for sensor nodes in a WSN to communicate. Hence, WSN routing needs to consider ad-hoc network routing issues such as connectivity, coverage and communication reliability/delay. However, routing in WSN differs with ad-hoc network due to the sensor nodes resource restrictions and WSN characteristics. This means that WSN routing would need to consider additional issues that are listed as below [Akkaya and Younis, 2005], [Al-Karaki and Kamal, 2004]:

1. Conventional IP-based routing protocols (that are used usually in ad-hoc networks) cannot be utilised in WSN as the overhead of maintaining a global ad-

dressing scheme is high [Al-Karaki and Kamal, 2004]. In other words, it is extremely expensive to use a global addressing scheme in WSN due to three reasons: first, the nodes are resource constrained (limited communication, computation, storage and power resources), second, they are usually distributed randomly in harsh environments, and third, the network may be broadly deployed with a large number of nodes.

2. Although flooding and gossiping routing protocols [Akkaya and Younis, 2005] can be utilised in ad-hoc networks to transmit data (because of their simplicity in implementation), they are not recommended to be used in WSN because of their high cost of performance. These protocols waste network resources by transmitting unnecessary or redundant copies of sensor data that may be measured from overlapping areas.

3. Hardware limitations in sensor nodes require accurate resource (especially energy) management in WSN. Owing to this, and due to the fact that routing in WSN (especially link availability/reliability) highly depends on the residual energy level at nodes, WSNs need to conserve energy by utilising lightweight and efficient energy use routing protocols.

4. In contrast to ad-hoc networks, routing in WSN is application dependent. This means that the sensor nodes establish the paths to collect and report the data by considering the scenario that is defined by the data consumer. For example, source nodes need to forward their data packets from source regions to a set of intermediate aggregators for in-network aggregations in data aggregation applications, whereas they need to forward query packets to respective data source nodes in Distributed Sensor Database System (DSDS) applications [Bonnet et al., 2001].

5. WSN routing focuses on forwarding data packets from data regions to data consumer access points (sinks). Hence, they need to be location-aware and/or data-centric to save network resources and avoid blind routing via random nodes.

According to the differences, it is concluded that ad-hoc routing protocols are not suitable for WSNs. A lack of global addressing, network density and resource constraints are the key issues that motivate researchers to design and propose novel and lightweight routing protocols dedicated to WSN. The routing protocols may consider the key features of ad-hoc routing (i.e location or energy awareness) due to the similarities. However, they need to utilise efficient mechanisms/techniques (i.e data centric routings) due to the distinctive characteristics of WSN such as lack of global addressing scheme and resource limitations.

2.4.2 WSN Routing Design Issues

A set of distinct factors need to be considered when designing WSN routing protocols. They are rooted in the network architecture, routing operations and/or sensor nodes capacities [Akkaya and Younis, 2005], [Yang and Mohammed, 2010], [Al-Karaki and Kamal, 2004]. The key ones are explained below:

- **Network architecture:** it has a significant impact on routing to discover and establish routes that are used to forward packets. Data packets need to be forwarded from source nodes to their higher levels of hierarchy such as the leader/cluster-head in hierarchical WSN, whereas they are forwarded to the sink either directly or indirectly in a flat network.

- **Node placement:** node placement is rooted in the network applications and/or the consumer requirements and has the potential to influence routing connectivity and coverage. Sensor nodes can be placed in two schemes: deterministic and non-deterministic. In the former, the sensor nodes are manually placed in the field in which the network traffic is forwarded through pre-determined routes. In the latter, the nodes are randomly scattered in the field wherein paths are dynamically formed with respect to connectivity and coverage.

- **Energy:** routing performance highly depends on energy consumption in WSN. As the required power for wireless communications is correlated to distance, forwarding network traffic through shortest paths is highly desirable to conserve energy.

- **Data delivery model:** routing is influenced by the data delivery models which are continuous, query-driven, event-driven or hybrid. For example, single path routing is not recommended in continuous data delivery (i.e habitat monitoring) as transmitting all the packets continuously through a single/same path can drain the energy of nodes being used (bottleneck). Owing to this, multi-path or hierarchical routing protocols are utilised to forward the network traffic through a set of variant paths or intermediate nodes that are able to eliminate redundant data.

- **Node capabilities:** nodes capability and functionality influences routing design and performance. For example, data packets are forwarded from the source nodes to the sink through intermediate nodes that have particular abilities such as in-network data aggregation.

- **MAC protocol design:** MAC protocols affect the routing performance as they are responsible for wireless link availability. They influence link availability which results in changing communication connectivity and/or coverage. MAC protocols are used in WSN to enhance energy efficiency by preventing sensor nodes to consume energy for idle-listening. They allow nodes to wake up when they need to send or receive network packets and then go to sleep if they have nothing to do.

- **Data aggregation:** as source nodes may forward a large amounts of redundant data measured from overlapped areas and/or similar events, data aggregation techniques are used by routing protocols to reduce the size and/or the number of (similar and/or redundant) data packets.

2.4.3 Classification of WSN Routing Algorithms

A number of routing protocols are designed and proposed for WSNs in which only a small number have been intensively studied and simulated, and even fewer have been implemented in the real world. Some of them have been proposed to work for specific scenarios and/or conditions, whereas others are more general and have been designed to work for a boarder set of applications and scenarios. WSN routing protocols are classified according to various criteria that are rooted in the network applications, architecture, and routing functionality. The classifications highlight the distinct features of routing protocols (i.e the type of routing information which is required to acquire and maintain, and proactivity and/or reactivity of routing information collection). It helps researchers to compare routing protocols to find their similarity, difference and/or superiority.

WSN routing protocols are classified in two levels based on network architecture and routing protocol operation. First, the routing protocols are classified according to the network architecture as flat and hierarchical. Second focuses on routing operation factors such as the data delivery model, MAC protocol design, QoS parameters, energy consumption and node capabilities. Multi path, query-based, negotiation-based, QoS-based and/or coherent-based routing protocols are the key categories of this level. According to figure 2-1, network architecture classification is the key one that routing operation categories are considered as parts of. A brief discussion is provided to explain each WSN routing category in which the routing protocols that are mature, famous and/or simulated or implemented in the real world are presented and explained. This section does not provide a statical analysis of the routing protocols, but it explains and highlights a set of distinctive techniques, features and schemes that are used in the

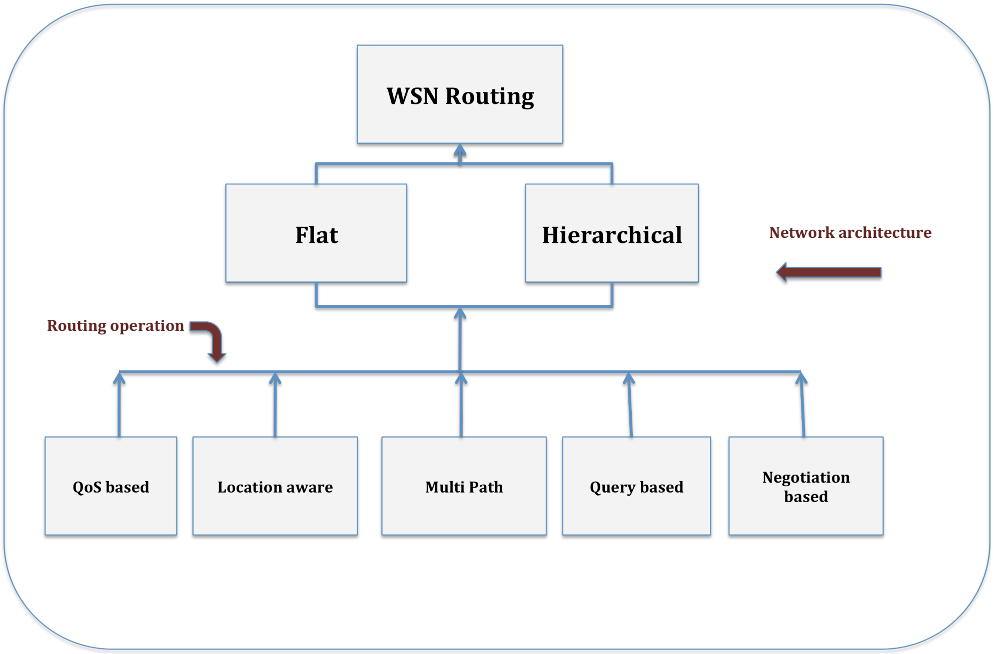


Figure 2-1: WSN Routing Classifications

introduced protocols. The subsequent chapters will provide statical analyses of selected routing protocols to highlight and compare their strengths and weaknesses.

2.4.4 Flat Routing Protocols

Flat routing focuses on forwarding data packets over a flat infrastructure in which all nodes play the same role. The nodes usually have similar capabilities and perform similar tasks to collect data and transmit the network traffic. In flat routing, the source nodes measure and forward environmental data to the sink via relay nodes. Flat routing is extremely popular in WSN due to its ease of use and low cost of deployment. The nodes do not need to pay extra cost to establish or maintain communication infrastructures such as grid, tree and/or clusters to forward network traffic. Moreover, data packets are forwarded from source regions to the sink using intermediate nodes that do not need to use any specific communication patterns (i.e intra/inter cluster). Each intermediate node that receives the packets will proactively/reactively establish either a single or multi hop path to forward data packets to the sink.

Flooding and gossiping [Akkaya and Younis, 2005] are two classical routing protocols that forward data packets to the sink. The former focuses on broadcasting data packets at each node. Data packets are broadcasted until they are reached by the sink. Although flooding is easy to implement, it has a number of drawbacks, specifically message conflict/loss/congestion, overlap data reports and quick network energy drain. Indeed, transmitting duplicated messages and reporting data packets that are measured from overlapped data regions via flooding algorithms significantly increases network resource consumption, traffic, collision and delays. Gossiping is proposed to resolve the drawbacks by forwarding the data packets to a selected set of intermediate nodes (multicast) instead of all neighbour nodes (broadcast). In gossiping, each source node selects a set of its single-hop neighbours (randomly or based on specific features like distance to sink) to forward data packets. Each intermediate node performs a similar algorithm to forward data packets until they are received by the sink. For this reason, the number of transmitted data packets will decrease. It results in reducing network resource consumption and traffic.

Sensor Protocols for Information via Negotiation (SPIN) [Kulik et al., 1999] is a negotiation-based data-centric (DC) routing protocol that uses meta data instead of original data to establish the routes over a flat WSN. First, each source node advertises its data (using a high-level data indicator message) in its single-hop neighbourhood. Each neighbour node that is interested in collecting and reporting data replies back by sending a request packet. Finally, the source node sends the original data packet to any of its neighbours who already asked for. The intermediate nodes perform a similar scheme to forward data packets until the sink receives them. As the author claims, SPIN outperforms flooding and gossiping in terms of network resource (mainly energy and bandwidth) consumption, traffic and collision because it reduces redundant transmissions. However, SPIN is not able to guarantee data delivery because of the utilising negotiation-based data transmission scheme to forward data packets. This means that source nodes do not forward data packets if they do not receive a request for data if there is a hole (disconnection) or request packets are lost. Moreover, SPIN can not be considered energy-efficient as the intermediate sensor nodes waste energy resources to stay available over long periods to receive data advertisements.

Directed DiFFusion (DDiFF) [Intanagonwiwat et al., 2000] is a query-driven routing protocol that utilises data naming technique to forward data packets. The objective of utilising the attribute-value is to reduce network resource consumption by eliminating unnecessary data processing and communication at the intermediate nodes. This means that a data packet is forwarded if it matches the query attribute-values. Data processing and communication is reduced as routing is performed only by the nodes that have interesting data for the sink to report or is able to establish a link to the source regions. First, the sink propagates data queries containing a set of attribute values (i.e data type, freshness ratio and/or geographical area) which describes the interesting data to collect. The queries are periodically propagated by the sink to check/refresh the availability of possible routes. Upon receiving the query packets, intermediate nodes record routing information of packets at their local tables to establish/reserve return links from the source nodes to the sink. The recorded routing information is used for in-network data aggregation when the data packets are transmitting from the source nodes through the return paths. The query propagation is repeated by intermediate nodes until they are received by the source nodes that have interesting data. As source nodes receive a number of similar queries that are forwarded through variant routes, they need to select the most optimal path to report data packets. In turn, they therefore consider a set of factors (like end-to-end delay or hop-count) to establish a low cost and/or delay links. The selected path (called gradient) is used to forward data packet to sink. Other possible routes (which are recorded at the intermediate nodes) are used as alternatives when the gradient fails. Data packets are forwarded by intermediate nodes that reside on the gradient until they are received by the sink. According to the simulation results in [Intanagonwiwat et al., 2000], Directed diffusion outperforms flooding and gossiping in terms of network resource consumption (energy) and delay. Energy consumption reduces as intermediate nodes use unicast instead of multicast or broadcast to transmit data packets, whereas delay decreases because of forwarding data packets via minimum hop count paths (gradients) from source regions to the sink. However, DDiFF is not particularly useful for continuous data collection as nodes need to consume a great deal of energy to transmit multiple queries and data packets. In other words, nodes (bottleneck) residing on the gradient consume a greater deal of energy to transmit queries and data packets which are frequently transmitted.

Gradient-Based Routing (GBR) [Schurgers and Srivastava, 2001] considers hop count between source nodes and the sink to route data packets. Each source node forwards a route request packet to establish a minimum hop count path to the sink. The request packets are forwarded by intermediate nodes until they are received by the sink. The minimum-hop-count path is selected as the node gradient. In turn, each node is assigned by a hop count value which is called node height. Hence, data packets are forwarded from the source nodes to the sink through the gradient. However, a number of gradients may be established if the route request packets are received by the sink via variant routes. GBR considers three schemes to select a path if multiple gradients are available: (1) stochastic scheme, which randomly selects a path when multiple gradients are established with same hop count, (2) energy-aware that considers residual energy level at each node residing on the path to forward data packets. A node height is set to a maximum level when its residual energy level drops under the threshold that is required to transmit data packets. Hence, source nodes stop forwarding data packets via the nodes to the sink. (3) stream-based scheme focuses on selecting a gradient to forward data packets from each source region. Data packets are classified based on data region, type and/or collection time. Then, the gradients are classified according to the data categories to forward data packets. This means that a gradient is used to forward data packets from each category (i.e data region) to uniformly balance network loads. GBR supports data aggregation to reduce the number/length of transmissions. For this reason, a set of nodes which are on multiple gradients are identified as intermediate aggregators to perform in-network data aggregation. Unlike DDiFF, GBR forwards data packets which are not desirable for the sink as it does not consider the sink queries. It results in increasing network traffic and energy consumption and consequently reducing network lifetime especially when the network works over long periods.

Energy Aware Routing (EAR) [Shah and Rabaey, 2002] establishes low energy use paths to forward the network traffic. This protocol utilises a class-based addressing scheme in which an ID is allocated to each node according to the node location and data type. EAR performs three phases namely set-up, data transmission and route maintenance. During set-up, the destination node (sink) floods a query/interest packet which is assigned a cost value initiated to zero. The interest packets are forwarded to measure the cost of established paths to the source regions. The cost value is measured according to equation 2.1. Upon receiving the interest message (from node A for example), each intermediate node (node B) adds the cost of the last link (Metric (A, B)) to the message's cost value. The link cost (Metric (A, B)) is measured according to the required energy to establish the link or the residual energy level at the nodes. Low cost links are maintained at Forwarding Tables (FT) and links with great cost values are discarded. Besides, a probability value is allocated to each link according to equation 2.2. The probability value is used to select the most optimal link in terms of energy consumption. Then, the interest message is updated according to the calculated cost value and then it is forwarded to the single hop neighbours which are closer to the source regions and farther from the sink than the sender. The class-based addressing scheme is used to find the neighbours which reside closer/farther to source nodes and/or the sink.

$$Cost_{(A,B)} = Cost_{(A)} + Metric(A, B) \tag{2.1}$$

In data transmission phase, the source nodes estimate the average cost value which is required for a packet to reach the sink. A link is established to forward the data packet if the link to the next hop node has a same probability value as the sender. Data packets are forwarded in the same manner until they are reached by the sink. In the route maintenance phase, nodes utilise local flooding (using HELLO messages) to periodically refresh the established links. If the current link fails, the next path will be selected from the routing table (FT) to forward the data packets.

$$P_{(A,B)} = \frac{\frac{1}{Cost_{(A,B)}}}{\sum_{C \in FT_{(A)}} \frac{1}{Cost_{(A,C)}}} \tag{2.2}$$

EAR protocol has the potential to resolve the existing drawback of Directed diffusion. It discards the paths that are used frequently to avoid bottlenecks. As the residual energy level at the intermediate nodes is considered to calculate the path cost value, the nodes residing on the frequent used links get high cost. Thus, the links are discarded from FTs when the level of energy at a node falls below the allowed threshold. However, data collection delay is not considered in EAR as the paths are established according to the energy cost. Moreover, utilising flooding to refresh the established links increases network resource consumption and congestion.

Minimum Cost Forwarding Algorithm (MCFA) [Ye et al., 2001] establishes multiple routes from data regions to the sink in which the path with the minimum cost (hop count) is selected to forwarded data packets. In contrast to EAR [Shah and Rabaey, 2002], MCFA does not use any specific addressing scheme (i.e class-based) to forward the network traffic. Prior to network deployment, each node is assigned by a cost value that is initiated to infinity. The cost value is updated upon a cost packet is received by a node. The cost packet is used to let the nodes know how many hops they are away from the sink. The sink initiates the hop count value to zero and then broadcasts the cost packet. The hop count value is incremented by one at each node which receives. The cost value of each node is updated to the new hop count value and then the cost message is forwarded for the next hop nodes. It is repeated until all nodes update their cost value according to the cost message hop count. Then, data packets are assigned by the cost value at source nodes and then they are forwarded via intermediate nodes that have a smaller cost value (are closer to the sink) until the sink is reached. Although MCFA is simpler than EAR to implement (the cost and probability functions are not required), it increases network traffic as the number of transmitted cost packets increases. This means that a large number of redundant cost packets is blindly re-transmitted by the intermediate nodes. In consequence, the message conflict ratio will increase especially when the network deployed is dense. MCFA has been extended in [Henderson and Tron, 2006] by utilising a technique that allows the intermediate sensor nodes to wait for a period before re-transmitting the cost packets. Using this technique, the receiver would be able to consider/aggregate a number of cost packets before re-transmitting them. It would result in reductions to

Table 2.8: Flat WSN routing protocols

Protocol	**Mechanism**	**Key advantage**	**Key drawback**	**communication**	**Routing metric**
Gossiping	Selective broadcasters	ease of implementation	1- high routing cost/delay 2- message collision/congestion	multicast	fresh routes
SPIN	Negotiation-based	reducing the traffic of query broadcasts	data delivery not guaranteed	multi/unicast	data centric
Directed diffusion	Query-driven	1- reducing energy/delay of query broadcasts	bottlenecks	unicast	1- minimum hop count 2- data centric
GBR	Multiple paths	1- minimising delay 2- increasing path availability	increasing energy cost	uni/multicast	shortest path (hop count)
EAR	Energy-aware	1- reducing energy 2- avoiding bottlenecks	1- delay not considered 2- flooding to refresh routes	multi/broadcast	minimum energy cost
MCFA	Multiple paths	easy setup	increasing message conflicts	uni/multicast	minimum hop count
Rumor	Query-driven	reducing query broadcasts	agents TTL	uni/multicast	data centric

network traffic and consequently decrease the energy consumption.

Rumor [Braginsky and Estrin, 2002] utilises an event flooding technique to reduce overhead of the sink interest propagation. Unlike directed diffusion in which the sink queries are flooded into the network (without any geographical assumption), Rumor forwards the queries only to the regions that have interesting data for the sink. In this protocol, source nodes generate and forward long-live packets, which are called advertisement agents, to inform their neighbourhood about their available data. Each node that receives an agent, updates its routing table using the information of the agent routing table that shows an available route to a source node. Using the information, the queries are not flooded to the source regions but they are forwarded through the paths to the interesting events/source nodes. Rumor has the potential to reduce the overhead of query propagation in comparison to flooding by avoiding to broadcast the sink queries. However, the energy efficiency of this protocol is increased in the case of frequent events. That is, a greater number of forwarding agents are required to advertise the events forwarded. Hence, network resource consumption would increase. Moreover, defining agent Time To Live (TTL) is a challenging issue in Rumor. The overhead of Rumor will increase if TTL is great as the agents move over longer distance paths. On the other hand, the overhead of query propagation will be increased when TTL is assumed to be really small because the agents can cover smaller parts of network to advertise data. Thus, the nodes that do not receive agents may need to use blind flooding to propagate the sink interests.

The introduced flat routing protocols are highlighted and compared in Table2.8.

2.4.5 Hierarchical Routing Protocols

Hierarchical WSN routing establishes a hierarchical infrastructure to forward data packets from event regions to the sink(s). The hierarchical infrastructures (i.e tree, cluster, chain and/or grid) are usually established to enhance routing performance. Hierarchical routing has the potential to offer following benefits:

1. **In-network data aggregation:** hierarchical routing may allow sensor nodes to perform in-network data aggregation. Data packets are transmitted from source nodes to higher levels of hierarchy (i.e leader and/or cluster-heads) to aggregate. It leads to reduce the number of transmissions and network congestions. In other words, message collision is reduced in hierarchical routing as a fewer number of data packets are transmitted over limited wireless channels. Clustering, for example, is a technique which is commonly used to establish hierarchical infrastructures in WSN. In a clustered WSN, data packets usually are transmitted from source nodes (cluster members) to cluster-heads to collect, aggregate and/or transmit. Aggregated results are reported by cluster-heads to sink via either direct or indirect links.

2. **Increasing the message delivery ratio:** the probability of message failure/collision would be reduced in hierarchical routing as the network traffic decreases. In hierarchical routing, a single or multiple nodes are in charge of providing communication services for a group of nodes. Owing to this, each source node does not need to individually establish a path to the sink to forward its data. It reduces the number of nodes which try to access the wireless channels and the number of forwarding data packets. In consequence, message collision/failure is reduced, resulting in increasing the message delivery ratio.

3. **Fair channel allocation:** wireless communication channels can be efficiently managed in hierarchical network as contention-free scheduling is supported. Contention-free scheduling allocates wireless channel according to the nodes hierarchy level or location in advance of communications. It has the potential to increase the fairness of channel allocation and consequently decrease packet collisions as compared to contention-based scheduling which is increasingly used in flat networks.

4. **Uniform energy dissipation:** using hierarchical routing, network loads are processed in a distributed manner instead of centralised. Network traffic is hierarchically forwarded to a set of intermediate gateways to process, relay and/or analysis. It leads to balance network loads resulting in increased network lifetime. On the other hand, the nodes which forward data packets from the source regions to the sink become bottlenecks and consequently consume energy in a non-uniform manner if flat routing is used.

5. **Routing delay reduction:** hierarchical routing has the potential to reduce communication delay as data packets are forwarded through parallel hierarchi-

cal links. Besides, receive (queueing) and access delays [Ardakani et al., 2014] are reduced due to decreasing the number of messages and network traffic in hierarchical routing.

The cost of hierarchical infrastructure establishment and maintenance is a drawback of hierarchical routing. Hierarchical routing periodically or frequently consumes network resources to maintain the infrastructure connected. In other words, nodes need to consume network resources to establish, update and/or re-create the infrastructures according to the network topology changes. Hence, energy conservation is the key issue which the researchers/engineers should consider to design hierarchical routing protocols in WSN.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [Heinzelman et al., 2000] is a cluster-based routing algorithm that supports data aggregation. LEACH has two phases: setup and steady-state. The former focuses on network clustering, whereas the latter routes data packets from source nodes to sink. The cluster-heads (CHs) are periodically selected based on a distributed random algorithm in which each cluster member may become a CH for a particular round according to a probability value (P). The probability value allows a cluster member to become a CH for 1/P round. In other words, there is no chance for a node to become again CH up to P next rounds. The source nodes collect and transmit data samples to cluster-heads using TDMA (Time Division Multiple Access) [Mao et al., 2007] to avoid intra-cluster collisions. Cluster-heads collect and aggregate data samples and then transmit the results to the sink. CDMA (Code Division Multiple Access) [Ares et al., 2007] is used by CHs to avoid inter-cluster interference. LEACH reduces energy consumption due to forwarding data packets in unicast instead of multi and/or broadcast. However, there are three drawbacks in (original)LEACH: (1) periodical CH selection to replace low battery CHs with new ones would extremely increase network energy consumption. (2) establishing single-hop inter and/or intra cluster links between the sink, cluster-heads and cluster members to forward data is not feasible for WSNs which are deployed in large areas. (3) non-balance CHs distribution and uncertainty in cluster count and size . For this reason, a set of modified version of LEACH are proposed aiming to resolve the drawbacks. A Two Level LEACH (TL-LEACH) [Loscr et al., 2005] resolves inter/intra cluster single hop communications by establishing a two-level clustered infrastructure. They are called primary and secondary clusters. The secondary cluster-heads collect data from the source nodes and then transmit the aggregated results to the primary ones to report to the sink. Multi-hop Routing with LEACH (MR-LEACH) [Farooq et al., 2010] extends TL-LEACH by providing multi-hop paths between the CHs to transmit data packets to the sink. Each cluster head would relay data packets that are forwarded from the CHs residing in lower hierarchy level via multi-hop paths to sink. Energy aware LEACH (E-LEACH) [Xiangning and Yulin, 2007] initially selects CHs similar to original LEACH (randomly) and then utilises residual energy at each nodes to select the CHs for next rounds. LEACH-centralised (LEACH-C) [Heinzelman et al., 2002] uses the sink as a centralised point to create the clusters in an optimal way. The sensor nodes forward the required clustering information such as location, residual energy and/or connectivity degree to the sink during the set-up phase. The sink proactively forms a set of balance clusters in terms of energy, coverage and connectivity and then allocates the roles (i.e CH and/or cluster member) to the nodes. The overhead of collecting clustering information at the sink to form the clusters is a drawback of LEACH-C. Vice-CH LEACH (V-LEACH) [Yassein et al., 2009] selects a vice-cluster head at each cluster to handle cluster communications is the case of CH failure. The CHs in LEACH may fail quicker (due to running out of energy) than cluster members as they usually need to perform a greater number of communication/computation tasks. For this reason, the vice-cluster head would stay in the cluster to cover CH duty if it fails. It would result in enhancing the network lifetime. LEACH Fuzzy Logic (LEACH-FL) [Ran et al., 2010] utilises fuzzy logics based on three metrics: residual energy level, density and distance from sink to select CHs. The author claims LEACH-FL has the potential to reduce energy consumption of CH selection and consequently enhance network lifetime.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [Lindsey et al., 2002] hierarchically forms a chain-based infrastructure to route data packets. It selects a set of nodes as leader nodes according to residual energy level and/or location information to collect, aggregate and transmit data samples. Data packets are forwarded from source nodes to the next hop nodes if they are closer to the leaders using a greedy algorithm. Each node aggregates the received data with its own and then transmits the result until the leader node receives. The leader nodes are responsible for reporting the results to the sink. If the leader node fails, sensor nodes leave the chain to construct a new one with a new leader. The difference of PEGASIS and LEACH is that the sensor nodes do not need to periodically pay the clustering cost to re-cluster the network. Due to this, and according to the simulation results in [Lindsey et al., 2002], PEGASIS outperforms LEACH in terms of network lifetime. However, the overhead of leader selection is increased in PEGASIS when the network works over long period. The sensor nodes will require a dynamic topology adjustment to collect information (i.e residual energy which changes over time) that are required to select or re-select the leaders. Moreover, the leader becomes bottleneck if data samples are frequently transmitted to the sink. Data collection delay also is increased in PEGASIS due to the multi-hop transmissions (with variant hop count) from source nodes to the sink. Hierarchical-PEGASIS [Lindsey et al., 2002] is proposed by the same author aiming to resolve the delay drawback. It reduces delay using parallel transmissions from the source nodes to the sink. It uses two techniques to provide parallel communications: signal coding (e.g CDMA) and transmitting spatial separated data. (1) signal coding: the nodes construct a tree of chains which is rooted in the sink. Using this tree, the nodes at each level transmit data packets to their leader in parallel. Each level of transmissions is coded by CDMA that allows collision-free parallel communications. (2) transmitting spatial separated data: this technique allows the nodes that physically reside close to each other to transmit data packets to the leaders at each round. In other words, it allows the source nodes to be grouped spatially and then group members independently transmit data samples to the leader.

Hybrid Energy-Efficient Distributed Clustering (HEED) [Younis and Fahmy, 2004] utilises a rotation-based clustering algorithm by considering four objectives: (1) utilising a finite number of iterations for cluster-head selection to reduce clustering/re-clustering overhead, (2) minimising network traffic by reducing the number of control packets that are forwarded during re-clustering procedure, (3) forming compact and well-distributed clusters by considering cluster-head connectivity degree, (4) uniform network load distribution over clusters to prolong network lifetime. HEED considers two metrics to select cluster-heads and form the clusters: residual energy (primary) and node connectivity degree (secondary). It forms the cluster by selecting the nodes that have sufficient energy and are able to dominate a specific number of nodes as CHs. It would result in forming well-distributed clusters in the network. The author claims HEED outperforms LEACH in terms of energy consumption due to reducing the number of re-clustering rotations. The clusters are re-formed according to a finite iteration number which is managed by the network consumer. In addition, the CH lifetime would be increased in HEED as CHs are selected in an energy-aware manner. On the other hand, the CHs have a higher chance to fail in LEACH (especially when network works over long period) as they are selected according to the probability value and without considering residual energy. However, finding an efficient number of clustering rotation is a challenging issue in HEED. It needs to be tuned according to the application requirements, network size and density. The network resource consumption is increased if the clusters are frequently formed, whereas bottlenecks and/or cluster-head failures (due to running out of energy) are increased if clusters are rarely formed.

Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN) [Manjeshwar and Agrawal, 2001] utilises a reactive data-centric approach to cluster the network in which data samples are reported in a time-sensitive manner. The source nodes are grouped into a set of clusters based on their available data samples. They are assigned by two parameters: Hard Threshold(HT) and Soft Threshold (ST). HT is a threshold value of sensed data samples that shows new type/value data samples. ST is the acceptable change rate of similar data samples. It leads to report data in the case of sudden changes. According to the threshold values, a data sample is transmitted to a CH if it is beyond HT and/or greater than ST. The CHs also consider the threshold values to report collected/aggregated data to the sink. TEEN allows a fewer number of source nodes (instead of all) to transmit data according to changes that are interesting for the data consumer. It reduces data collection delay as the number of source nodes which try to access the wireless channels to forward data packets is reduced. However, TEEN is unsuitable for periodic data collection as it returns no result until data values are changed and the thresholds are reached. Adaptive Periodic Threshold Sensitive Energy Efficient Sensor Network Protocol (APTEEN) [Manjeshwar and Agrawal, 2002] is proposed by the same author to resolve the existing drawback. APTEEN is able to support three type of queries: (1) historical (by analysing past data values), (2) one-time (collecting the data once as a network snapshot), (3) persistent (monitoring the environmental events over a period continuously). That is, each CH sends a message containing four parameters to its CMs: (1) Attributes (A): it shows the elements of data samples that data consumer is interested in. Using attributes, source nodes avoid transmitting data samples that are not interesting to be collected. (2) Thresholds (HT and ST similar to TEEN), (3) Scheduling metric: it assigns a time slot to each CM to provide intra-cluster scheduling, (4) Count Time (CT): it is the maximum allowed period between two data reports. Similar to TEEN, source nodes transmit data samples when the values are beyond HT and/or greater than ST. However, source nodes re-sense and/or re-transmit data samples if CT counter is expired. Hence, CT enhances data collection flexibility when data consumer needs periodic data gathering. Assigning the threshold values to the nodes is a challenging issue for both TEEN and APTEEN algorithms. As the values are rooted in the application/consumer requirements, it reduces network flexibility if they are allocated to the nodes in prior of deployment. On the other hand, threshold value allocation consumes network resources if the values are frequently flooded from the sink to the CHs especially in dense networks.

Table 2.9 highlights and compares the key features of introduced hierarchical routing protocols.

2.4.6 WSN Routing Operation Classification

This section classifies the routing protocols according to the functions such as location-aware, QoS-based, Multi-path establishment, negotiation-based and/or coherent-based

Table 2.9: Hierarchical WSN routing protocols

Protocol	Architecture	Key advantage	Key drawback	Mechanism	Routing metric
LEACH	AC cluster	1- reducing message collision 2- increasing lifetime	1- random and non-balance clusters 2- not defined number of iterations	rotated clustering with infinite iterations	fresh routes
PEGASIS	Chain	reducing cost of infrastructure establishment/maintenance	1- bottlenecks 2- increasing delay	greedy distance (hop count) to sink	shortest paths
HEED	AC cluster	1- finite number of iterations 2- well-distributed clusters	tuning the number of re-clustering rotations	rotated clustering with constant iterations	fresh routes
TEEN	DC cluster	reducing redundant data reports	1- periodical report is not supported 2- threshold values allocation	data-aware clustering	data centric
ATEEN	DC cluster	flexible for different types of data reports	threshold values allocation	data-aware clustering	data centric

that are used during routing procedure. Routing operation classification resides on the top of network architecture. This means that routing protocols are classified according to the routing functions that are used over either flat or hierarchical infrastructures. The key categories are explained in the following sections.

2.4.6.1 Location-based Routing Protocols

WSN location-based routing is addressed by means of node locations to measure path distance, forward network traffic and/or find destinations. The benefits of utilising location-based routing in WSN are energy conservation and reducing network traffic. Using location-aware routing, the required routing information is acquired from the location information provider (i.e GPS) to establish the paths. Hence, the nodes do not need to broadcast frequent routing messages to heuristically establish the routes. It reduces network traffic and resource consumption to a great extent. Besides, the node which reside out of the communication regions/paths would be able to turn off their communication modules to save energy without influencing the route connectivity and/or network coverage. On the other hand, nodes need to re-transmit routing packets/requests until the destination/sink is heuristically reached in non location-aided routing due to lack of a global addressing scheme to find the destinations.

General Positioning System (GPS) and Received Signal Strength Indicator (RSSI) are two commonly used techniques in WSN to provide location information. In the former, the sensor nodes are equipped by a local and low-power use GPS to get the location information from the satellite. However, the communication cost between GPS and satellite and the availability of GPS connections to the satellite are two challenging issues for GPS equipped WSNs. RSSI technique allows the nodes to measure the distance to the destination using strength of received signals. The drawback of this technique is that exchanging messages between the nodes to measure distance increases routing overhead especially when the network is dense and large.

Geographic Adaptive Fidelity (GAF) [Xu et al., 2001] establishes a virtual grid infrastructure according to the location information that are provided by GPS. The grid square-width is equivalent to nodes radio range. Local communications in each square are managed by a node (called Grid Master) that have highest level of residual energy amongst all nodes in its local vicinity. To select Gird Masters (GMs), the sensor nodes are ranked based on residual energy level in a decentralised manner at each square. GAF is energy aware and assumes that each node can be in three statuses: sleep, discovery or active. A node which is in sleep mode, turns off its communication module to save energy as it has nothing to do. Discovery nodes collect location information to discover their local vicinities, whereas active nodes turn on their radio modules to transmit network traffic. After vicinity discovery and GM selection, square members go to sleep for a while until they are awake by the respected GM. The GMs stay on duty to handle messages that may be forwarded by source or GM nodes. GAF maximises network lifetime as it has the potential to reduce the number of active nodes in network without influencing network coverage and connectivity. However, the availability of GMs to manage intra or inter-square communications is the a drawback of GAF. GMs are selected proactively based on the energy level which changes over time. As GMs are in charge of managing square communications, they become bottleneck and fail (due to running out of energy) quicker than other nodes if the network is deployed to work over long periods. It results in disconnecting square members from the networks. On the other hand, frequent GM selection is expensive for WSNs as it needs local message broadcast to exchange the battery level information at each square.

Geographic and Energy-Aware Routing (GEAR) [Yu et al., 2001] uses location information and residual level of energy to heuristically establish routes over a flat network. It assumes that each node knows its location and is able to collect location information and remaining energy level of its single-hop neighbours. Each source node assigns two metrics called estimated and learning cost to the route request packet and then forwards it to the destination. The estimated cost is the amount of energy which is required to reach the destination. It is estimated at source nodes using a function of residual energy and distance to the target. The learned cost, is an updated value of the estimated cost which is dynamically measured at each node. Upon receiving the route request packet, learned cost value is updated according to the length of last traversed link and residual energy level at the node. Then, the packet is forwarded if the closest (minimum cost) single-hop neighbour node to the target region is found. This greedy algorithm is performed at each intermediate nodes until the destination is reached. In this case, the packet is forwarded through the destination region to target node using either a recursive geographic forwarding or restricted flooding technique. In recursive geographic forwarding, the region is divided into four parts. Four copies of

Table 2.10: WSN location-aided routing protocols

Features	**GAF**	**GEAR**
Mechanism	proactive location-aware zones	residual energy and location info.
Architecture	Hierarchical	Flat
Communication Type	limited broadcast	uni/multicast
Key advantage	1- reduced number of active node 2- reduced msg. conflicts	minimum cost paths
Key drawback	1- bottleneck at GMs 2- proactive GM selection based on energy	tuning the size of destination region
Routing metric	shortest/freshest path	shortest and minimum energy consumed links

the packet are separately created and forwarded to each part. This procedure is recursively repeated until the packet is reached by target node. In the restricted flooding, the packet is broadcasted to all the nodes which reside in target region. Although restricted flooding is simpler to implement, it increases energy consumption especially when the deployed network is dense. The size of target region is a challenging issue in GERA. The routing overhead would be increased when destination region is large, whereas the routing delay is increased if it is small. A greater number of network messages (copied or flooded) is forwarded to the target node when the destination region is large. It would result in increasing network traffic and consequently increasing energy consumption. On the other hand, a greater number of intermediate nodes would participate in routing procedure from the source nodes to the destination region to measure and update the path cost. It increases routing delay especially when the network deployed is large.

The presented location-aided routing protocols are highlighted and compared in Table 2.10.

2.4.6.2 Query-based routing protocol

A link is established in query-based routing if the consumer/sink requirements are met. The paths are usually established in a data centric manner according to the data consumer queries such as data type, location of interesting event and/or sensing time. They may be temporary or continuously used depending on the application and/or consumer requirements. The advantage of query-based routing is that source nodes do not waste network resources to establish random links or forward uninteresting data packets which are not required to be collected. However, query-based routing increases network resource consumption and/or delay when the sink frequently propagates the queries. The source nodes need to consume energy to establish a separate route for each query.

Constrained anisotropic diffusion routing (CADR) [Chu et al., 2002] aims to maximise data sample collection while minimising delay and bandwidth utilisation. CADR has two phases: first, Information-Driven Sensor Querying (IDSQ) which is responsible for forming a set of data centric regions comprising of source nodes that have desirable data to report. Second, is in process of forwarding data packets from source regions to the sink. Each data region is managed by a single or multiple leader nodes that stay in duty to collect data samples from the source nodes and forward the results to the sink. The leader nodes are selected in an incremental manner in which the number of collected data samples is maximised while delay is minimised. This means that IDSQ selects the nodes that have maximum connectivity degree to the desirable source nodes as the leaders. It allows the leader nodes collect maximum number of data samples through low delay (proactive) links. Then, sink queries are forwarded to the leader nodes at each data regions. Leader nodes that match the sink queries collect data samples from the connected source nodes and then forward the results to sink for further processing. The performance of CADR enhances when small data regions are formed in Event-Radius (ER) fashion in which source nodes are centralised. In other words, energy consumption and routing delay is reduced as integrated data regions (ER) allow the leader nodes to communicate to source nodes through low delay links that are likely single-hop. However, energy consumption and especially routing delay is increased when data regions are broadly formed in Random-Source (RS). In this case, sink queries need to be broadly propagated to cover the maximum number of leader nodes. It increases routing delay and energy consumption. Besides, data samples are delivered to the sink through variant and/or high delay routes as data regions are randomly distributed throughout the network.

Active Query Forwarding in Sensor Networks (ACQUIRE) [Sadagopan et al., 2005] utilises a data-centric query forwarding scheme for data collection. The sink forwards data queries comprising of required data elements that should be reported and collected. The queries are propagated by any intermediate node to a D-hop neighbourhood until they are received by the source nodes. When a sink query is received, a backward path is established to forward data packets if the receiver has the requested data at its local cache. Otherwise, the node forwards the queries to source nodes that are not father than D-hop away. The source nodes that receive the queries, forward their up-to-date data samples via the same path to the last query broadcaster. Each query broadcaster collects and aggregate data samples form its D-hop neighbourhood and then similarly forward the aggregated results to its last query broadcaster until the sink is reached. The value of D is a challenging issue in ACQUIRE. Network resource consumption is increased if D is defined large. For example, the cost of ACQUIRE

Table 2.11: WSN query-based routing protocols

Features	**CADR**	**ACQUIRE**
Mechanism	DC leader selection	DC query forwarding in D-hop
Architecture	Hierarchical	Flat
Communication Type	limited broadcast	limited broadcast
Key advantage	1- reduced delay 2- enhanced data collection quality	avoids to forward redundant data packets
Key drawback	high cost/delay for RS event sources	tuning the value of D
Routing metric	DC links	DC links

equals to flooding if D is defined as network diameter. On the other hand, the queries need to be propagated by a great number of query broadcaster to cover whole the network if D is defined small. It increases network resource consumption (especially in dense network) as the number of broadcasting messages is increased.

The explained query-based routing protocols are compared in Table 2.11.

2.4.6.3 QoS-based routing protocol

QoS-based routing forwards network packets through paths which are established according to the data consumer quality of service requirements such as delay, reliability, fault tolerance and/or bandwidth utilisation. A QoS route is established when the required QoS metrics are guaranteed and/or provided. However, QoS-based routing avoids to consider other parameters such as energy consumption which is vital in WSN. In other words, forwarding data packets according to QoS metrics would result in reductions of energy efficiency and network lifetime.

SPEED routing protocol [He et al., 2003] forwards data packets through links in which required end-to-end delay is guaranteed. It assumes that each wireless communication link transmits data packets with a certain speed. According to it, each link is proactively labelled by the communication speed and link length. Data packet delivery delay on each link is estimated by dividing link length over the communication speed. Each node discovers its single-hop neighbourhood and then establishes the local links using a geographic forwarding technique namely Stateless Non-deterministic Geographic Forwarding (SNGF). SNGF has four steps: (1) Beacon exchanges to collect local vicinity information at each node. It results in establishing/discovery of multiple local paths in each single-hop neighbourhood, (2) Delay estimation to calculate delay since a packet is transmitted until an acknowledgement message is received through the same link. It is used to measure packet speed at each node to select the links which meets the speed requirements, (3) Neighbourhood feedback ratio that discards the links which cannot provide desired packet delivery delay and/or speed. Source nodes find single-hop neighbours which are not able to establish communication links according to the QoS requirements. They discard the unsuitable links and forward data packets only through the links which are able to guarantee QoS requirements. (4) Back-pressure re-routing is used to detect network congestion when a the packet is not delivered. It aims to find next available links that are able to meet QoS requirements when the current link fails. The probability of packet failures/lose is decreased as QoS links are reliable enough to forward data packets. However, energy consumption increases in SPEED due to local communications, neighbour feedbacks and re-routing the packets especially when the network deployed is dense.

Energy-aware QoS routing protocol [Akkaya and Younis, 2003] forwards data packets through energy efficient links in which the required end-to-end delay is met. To provide energy efficiency and communication reliability, a cost value is assigned to each link. The cost value is a function of node energy residual level, required energy for forwarding a packet and link message error rate. The path with smallest cost value is selected using Dijkstra algorithm amongst all available links that meet the end-to-end delay requirements. To guarantee end-to-end delay, network packets are classified using a class-based queuing model into real-time and non-real-time categories. Each class is assigned by a bandwidth parameter that represents the amount of available bandwidth on each link. Bandwidth ratio parameter is set at each node prior to the network deployment. According to the queueing classes, a packets is transmitted through the reserved bandwidth according to network consumer priority/requirement. Initial bandwidth allocation prior to deployment is a drawback of this protocol. It does not provide flexible bandwidth sharing adjustment between realtime and non-realtime class of transmissions when the network works over long period. The priority of bandwidth categories, network topology, available bandwidth and node communication power may change over time.

Table 2.12 compares the introduced WSN QoS-based routing protocols.

2.4.6.4 Multi-path routing protocols

This class of routing focuses on establishing multiple paths (rather than a single one) between the source nodes and the sink. Establishing multiple paths enhances routing performance and reliability as alternative links would be available when the current one fails. Moreover, providing alternative paths can reduce the impact of network topology changes (i.e intermediate node/link failures) on connectivity, coverage and/or message delivery. However, multiple path discovery and/or establishment increases routing overhead. It results in increasing network resource consumption especially for

Table 2.12: WSN QoS-based routing protocols

Features	**SPEED**	**Energy-aware QoS**
Mechanism	proactive link labelling based on communication speed/delay	class-based bandwidth allocation to energy-aware links
Architecture	Flat	Flat
Communication Type	limited broadcast	limited broadcast
Key advantage	1- end-to-end delay is guaranteed 2- enhancing packet delivery	energy-aware with QoS distribution model
Key drawback	1- not energy-aware 2- extra cost for multiple paths	pre-deployment bandwidth allocation
Routing metric	minimum delay links (QoS)	low energy and delay links

large and/or dense networks. Sensor nodes would need to consume a great deal of resources to discover, establish and maintain multiple communication links. Directed diffusion [Intanagonwiwat et al., 2000] is an example of multi-path routing in which multiple paths are discovered and established. The links are ordered based on the energy efficiency in which the most energy efficient path is used to forward data. Directed diffusion does not maintain all discovered paths alive as it is extremely expensive in terms of network resource consumption. But, it recovers next path from the list of discovered links to become available on-demand when the primary one fails.

2.4.6.5 Negotiation-based routing protocol

Negotiation routing establishes the links between the source and destination nodes in a connection-based manner in which the sender and receiver nodes communicate to find any available/reliable link between in advance of forwarding data packets. The negotiations let both sides of communication know about available network and data resources at each other. This means that senders and receivers share the availability and reliability of their resources during negotiation procedure. Therefore, the sender nodes avoid transmitting uninteresting, redundant and out-of-date data packets as they are already aware of the receiver interests. Negotiation routing reduces the probability of message failures as it has the potential to ensure sender about the availability/reliability of the receiver/link in advance of the communications. According to these reasons, negotiation-based routing has the potential to enhance the reliability of routing in WSN. However, negotiation-based routing increases network traffic and communication overhead. It needs several rounds message passing (between the sender and receiver for each data transmissions) that increases the number of communications, network traffic and consequently network resource consumption. SPIN [Kulik et al., 1999] is an example of negotiation-based routing. The source nodes disseminate data advertisement packets to find the potential paths which have are able to forward data packets to the sink. The intermediate nodes that know such the paths (either in address or data centric) reply back the sender to forward data packets.

2.5 Data Aggregation routing in WSN

WSN routing is rooted in the network applications. Sensor nodes route collected data samples based on data consumer scenarios and/or requirements from the source regions to the sink. Data aggregation is one of WSN applications that needs routing to collect, aggregate and deliver data samples to the sink. Data aggregation routing has two schemes [Xu and Qi, 2008], [Biswas et al., 2008]: client/server and mobile agent. In the former, data packets are forwarded from source nodes to either intermediate aggregators or the sink to collect and/or aggregate according to network architecture (flat or hierarchical). In the latter, a single or multiple mobile agent(s) which is/are in charge of data aggregation move throughout the network to collect and aggregate data. Mobile agent routing scheme needs to provide paths between the data consumer access point (sink) and the event regions for mobile agents to move. In other words, the mobile agents need to migrate through the paths to capture and aggregate data samples at the source nodes and then return the results to sink.

2.5.1 Client/server data aggregation

Client/server data aggregation routing establishes the paths according to the network architecture that can be flat or hierarchical. In flat networks, the routes are established from the source nodes to sink in a convergent manner as all nodes play same roles. Apart from sink, intermediate nodes may perform in-network data aggregation if they receive multiple data packets. However, no node is particularly selected to perform data aggregation in flat networks. On the other hand, the nodes may play different roles such as network bridge, intermediate aggregator or data consumer access point in hierarchical networks. Hierarchical routes are usually established from the source nodes to the sink via intermediate nodes which are in process of data aggregation. The intermediate nodes hierarchically aggregate and forward data packets to the sink.

2.5.1.1 Flat Networks

Flat data aggregation routing establishes low-cost routes from the source nodes to the sink to forward data samples. Data packets are reactively aggregated at intermediate nodes while they are being transmitted. Flat data aggregation routes may be established either in address-centric (AC) or data-centric (DC). The former focuses on establishing and recognising the paths based on the address of nodes, whereas the latter considers data packet content and/or data availability at the intermediate nodes to establish the paths. DC routing is commonly used to provide in-network data aggregation as it forwards through the nodes that are aware of data content and able to perform in-network data aggregation.

Flat client/server data aggregation routing protocols are categorised into three classes [Rajagopalan and Varshney, 2006]:

1. Push data aggregation routing: the sink initially establishes a subscription link (shortest path) to the source nodes which have interesting data to report. Source nodes, which receive the subscription links, become available to report data packets through the same links to the sink. Data packets are forwarded by the source nodes until the subscription links are available. Push data aggregation routing constantly consumes network resources as source nodes would frequently transmit data packets as long as the subscription links are available. In addition, the source nodes waste network resources by transmitting either redundant or uninteresting data to the sink as data packets are continuously forwarded. SPIN protocol (Sensor Protocols for Information via Negotiation) [Kulik et al., 1999] is an example of push data aggregation routing.

2. Two-phase pull aggregation routing: the source nodes heuristically establish shortest paths to transmit data packets to sink upon receiving the queries. It improves the quality and/or accuracy of data collection because data packets are forwarded according to the sink queries and not randomly or periodically. However, several round trip communications to establish the paths consumes network resources especially when network deployed is dense and sink queries are frequently changed. Direct diffusion [Intanagonwiwat et al., 2000], for example, utilises two-phase pull mechanism to collect and transmit data to the sink for aggregation.

3. One-phase pull aggregation routing: the sink establishes a shortest path to the source nodes that meet queries and have interesting data to report. The overhead/complexity of this technique is significantly increased (especially in dense and large WSNs) as the sink should be ware of the interesting source nodes location and data to establish the shortest paths. An one-phase pull data aggregation routing protocol is proposed by [Krishnamachari and Heidemann, 2004] in which sink propagates interest packets to the network to establish shortest paths namely gradients to the source nodes. Each source node which receives query packets se-

lects the minimum delay path (minimum hop count) to forward data packet if the query requirement is met.

The drawbacks of flat data aggregation routing are: (1) establishing shortest path between each source nodes and the sink consumes network resources especially when the network deployed is large and dense, (2) message failure and network congestion is increased as the source nodes simultaneously try to access the limited wireless channels to forward data packets. Data transmissions in flat routing are not integrated and data samples are forward by any source node that meets the query requirement in parallel, (3) re-transmitting data packets (which are lost/collided due to conflict/congestion) increases energy consumption. (4) data packet failures reduce data aggregation accuracy and robustness as the number of collected data samples at the sink reduces, (5) transmitting data packets through paths that have variant end-to-end delays can increase data collection end-to-end delay and influence data freshness [Al-Karaki and Kamal, 2004]. This means that data collection delay can be variant as data packets are delivered to sink through different paths with variant hop count. It increases uncertainty in data collection end-to-end delay that results in reduction of data freshness.

2.5.1.2 Hierarchical Networks

Hierarchical data aggregation establishes a hierarchical infrastructure to collect, aggregate and transmit data packets from the source nodes to the sink. It has the potential to resolve flat data aggregation drawbacks. In hierarchical data aggregation, source nodes do not try to transmit data packets to the sink but they forward data packets to a set of intermediate nodes which are particularly selected to stay on duty of performing in-network aggregation. Hence, data packets get aggregated earlier in hierarchical networks as compared to flat wherein an intermediate node performs data aggregation if it receives multiple data packets. In fact, instead of any random node that reside on a joint path, intermediate aggregators hierarchically aggregate data samples and forward then the results to the sink. Owing to this, network traffic and congestion is reduced in hierarchical data aggregation routing as the number of forwarding data packets and/or relay nodes is decreased [Hu et al., 2005]. It results in reduction of message collisions/lost, data collection delay and increasing data collection accuracy.

A hierarchical infrastructure can be formed in either static or dynamic fashion. The nodes are allocated by the network roles mainly intermediate aggregators in advance of network deployment in the former, whereas they are reactively selected and the hierarchical infrastructure is dynamically formed in the latter. Due to WSN characteristics such as network topology change and/or random deployment, dynamic establishing of hierarchical infrastructure is more popular.

Static hierarchical client/server data aggregation is easy and low-cost to set up as the intermediate aggregators are selected in prior to network deployment. The nodes do not need to consume network resources to dynamically form the infrastructure. Stronger nodes, in the terms of having sufficient computation, communication and storage resources, are positioned in efficient locations in which most possible number of source nodes are covered for data collection. It is commonly used in laboratory experimental situations because of their relative ease of installation. It allows source nodes to transmit data packets to the intermediate aggregators through energy efficient and low delay links. However, this scheme of hierarchical client/server data aggregation is not applicable for most applications of WSNs because of network topology change and random deployment.

Dynamic hierarchical data aggregation infrastructure is reactively formed in an ad-hoc manner by considering the available resources and capabilities of nodes such as remaining energy and/or coverage degree. The network is partitioned into a set of hierarchy levels (i.e clusters) that are managed by either a single or multiple nodes (i.e leaders or cluster heads). The nodes can be selected using voting and/or probability algorithms [Alex et al., 2008]. They are responsible for collecting and aggregating data samples and forwarding the results to the sink. Node density, distribution pattern, connectivity and coverage degrees are the issues that need to be considered to establish dynamic hierarchical infrastructure.

There are a set of different techniques that are used to form hierarchical infrastructures in WSNs. Aggregation tree, clusters and chain are the most commonly used ones that are explained as below:

1. **Aggregation tree:** a tree infrastructure is formed in which data packets are hierarchically forwarded by the source nodes to the parent nodes to perform in-network data aggregation. The aggregated results are hierarchically forwarded by the parent nodes at each level until they are received by the sink. The objective of establishing the aggregation tree is to minimise energy network resource consumption and maximise number of collected data samples [Anisi et al., September 16-18, 2011]. In other words, the aggregation tree should be established with the maximum number of interesting source nodes in an energy efficient manner. TAG (a Tiny AGgregation service for ad-hoc sensor networks) [Madden et al., 2002] proposes a dynamic hierarchical data aggregation protocol that establishes a tree infrastructure to collect and aggregate environmental data. First, a level discovery message is broadcasted by sink to allocate a level number to each node. Each node that receives the level discovery messages increases the level value by one and then re-broadcasts the message for the next hop nodes. This procedure

is repeated until all nodes receive the level values. Data packets are forwarded from the source nodes to their upper level nodes (parents) to collect and aggregate. They are hierarchically forwarded through the tree infrastructure until the aggregated results are received by the sink. TINA (a scheme for Temporal coherency-aware In-Network Aggregation) [Sharaf et al., 2003] similarly establishes a tree infrastructure to collect and aggregate data samples. The difference of TAG and TiNA is that the latter utilises temporal coherency tolerances to reduce energy consumption over the aggregation tree. This means that the source nodes in TiNA do not transmit all the measured data, but they just forward data packets whose values differ with data which is already reported. For this reason, a new parameter is added to sink queries which is called "tct". It shows the consumer preference tolerance degree to report a data sample. Considering this parameter, a data sample is reported if it differs with the last reported value greater than "tct". According to the simulation results [Sharaf et al., 2003], the number of transmissions as well as energy consumption is reduced in TiNA as compared to TAG.

2. **Clustering:** this technique partitions the network into a set of groups named clusters. The clusters are formed based on the similarity of nodes according to a set of distinctive features like location, measured data and/or communication and computation behaviours. Each cluster consists of a set of Cluster Members (CMs) in which a single or multiple ones are selected as the cluster representatives. The cluster representatives are called Cluster Heads (CHs) and responsible for collecting and aggregating intra-cluster data samples. Aggregated results are hierarchically forwarded from CHs to sink via single or multi-hop paths. CLUstered Diffusion with Dynamic data Aggregation protocol (CLUDDA) [Rajagopalan and Varshney, 2006] diffuses the sink queries into a clustered network in which the CHs are in charge of performing in-network data aggregation. The queries contain the required information (i.e data types, aggregation function and/or data freshness) for data collection. Each CH that meets the requirements collects and aggregates intra-cluster data samples and then forward the result to the sink. CLUDDA allows the data consumer to partially collect and aggregate data samples from each region of network in which data is desirable. It reduces energy consumption as data aggregation is performed by only a selective set of CHs (instead of all CHs) that match the interest packet requirements. However, redundant and/or overlapped data collection is the key drawback of CLUDDA. It means that, CHs perform data aggregation on data samples which are measured from same and/or overlapped area. It results in increasing network resource consumption and data

redundancy. Clustering-based heuristic for Maximum Lifetime Data Aggregation protocol (CMLDA) [Dasgupta et al., 2003] resolves this drawback by partitioning the network into a set of non-overlapped clusters with a constant number of members (CMs).

3. **Aggregation chain:** the hierarchical infrastructure for data collection/aggregation is formed as a chain of the source nodes which have interesting data to report. The chain is usually rooted in a node which is called leader and is responsible for reporting the aggregated result of chain members to the sink. Data samples are hierarchically forwarded and aggregated from the source region to the leader nodes. The leader(s) collect and forward the aggregated result to the sink directly or indirectly. PEGASIS (Power-Efficient GAthering in Sensor Information Systems) [Lindsey et al., 2002] establishes a chain infrastructure for data aggregation.

There are two key drawbacks in hierarchical client/server data aggregation [Liu and Sinha, 2007]: (1) infrastructure establishment and maintenance cost: the overhead of establishing and maintaining the hierarchical infrastructure increases network resource consumption and reduce network lifetime. Sensor nodes need to consume a great deal of energy to establish or re-establish the hierarchical structure when network topology or density changes, (2) Leader/CH bottlenecks: computation and communication task loads usually focus on the intermediate aggregators (i.e leaders or CHs) than other nodes in hierarchical data aggregation. Hence, the aggregators nodes have a higher chance to fail (due to running out of energy) as they need to manage a great number of communication and computation tasks.

Table 2.13 highlights and compares the key features of flat and hierarchical data aggregation routings in WSN.

2.5.2 Mobile Agent Data Aggregation

Mobile agent data aggregation forwards either a single or multiple mobile agents (MAs) throughout a network to collect and aggregate data samples from source nodes. The objective of mobile agent data aggregation is to enhance accuracy and performance of data aggregation by resolving existing drawbacks in client/server model such as load balancing, energy consumption and/or scalability. This section briefly describes mobile agents structures, capabilities and benefits. A set of mobile agent routing protocols are provided to highlight routing issues and techniques that need to be considered in WSN data aggregation.

2.5.2.1 Mobile Agents: a Brief Overview

A Mobile Agent (MA) is a piece of software that has mobility to autonomously perform distributed computing tasks [Cao et al., 2007], [de Freitas, 2011]. There are two techniques to provide MA mobility: mobile code and remote objects [Bieszczad et al., 1998]. In the former, code migration is provided and managed by a software framework (i.e Telescript Development Environment) [Nwana, 1996], whereas the latter (i.e Aglets) focuses on Remote Object Invocation (ROI) that allows remote access to the information/object with respect to the system transparency [Lange et al., 1997].

Programmability is the key feature of mobile agent as compared to regular computer softwares. It provides the ability of collecting and processing information, and then performing the best fitted services for the consumer. Programmability has the potential to enhance the performance of computing systems in which the MAs are used to intelligently manage resources [Lez-Valenzuela et al., 2011]. The MAs have the ability to support programming in application, middleware and network layers according to the definition domain [Aiello et al., 2009]. In the application layer, MAs are usually used to enhance the flexibility/efficiency of application design by propagating the consumer requirements in an autonomous manner. In the middleware layer, the MAs can be used to enhance dynamicity of network services such as data aggregation and/or query-based information retrieval. They also are used to improve the intelligence of network layer services such as smart multi-hop routing.

2.5.2.2 Mobile Agent Structure and Benefits for Data Aggregation

The MAs can be programmed to perform data aggregation in WSNs. They move throughout the network to capture and aggregate data samples which need to be reported to the sink. A mobile agent usually consists of four elements: identification, itinerary, data space and method [Qi et al., 2001]. The identification maintains the general information of MA such as serial number and/or dispatcher's ID. Itinerary provides the migration information such as current location, traversed paths and/or destination address. Data space keeps the required and/or collected data (i.e aggregated result) during the MA migrations. Method provides the required code/function (i.e aggregation function) that is used by MAs during the migrations between the computing devices. As a result, the MAs would be able to visit the source nodes one by one using the itinerary information that can be provided proactively or reactively. They aggregate captured data at each node using the aggregation function. Aggregated results are maintained and/or updated at data space of MAs until they are delivered to the sink.

According to [Mpitziopoulos et al., 2009] and [Qi et al., 2001], the MA model of data aggregation offers nine advantages in comparison to client/server as below:

1. Utilising the MA technique decreases transmission rate in WSN. MA routing forwards the executable sink queries (MAs) to the source nodes to collect and combine data samples, whereas client/server routes a large amounts of raw data from the source nodes to the aggregators/sink. For example, let we assume that a set of particular photos taken by wireless camera sensors need to be collected. In client/server, camera sensors report all their photos to either sink or aggregators for aggregation and/or processing/analysis. On the other hand, a MA can be programmed to move throughout the network to collect the photos which are interesting. Hence, the number of transmissions is reduced in MA data aggregation.

2. Reducing the number of transmissions results in decreasing network resource consumption mainly energy.

3. MA data aggregation reduces network traffic and message collisions as the number of transmissions decreases.

4. Data collection delay decreases due to reducing network traffic which results in reductions of buffering (receive delay) and/or wireless channel access delays (access delay).

5. MA model has the potential to enhance scalability due to independency of data aggregation performance to the network size.

6. MA improves the extensibility of system as it has the potential to carry task-adaptive processes which extend the capabilities of the system.

7. MA has the ability to improve the stability of network as it is able to support off-line (asynchronous) message delivery.

8. MA model reduces the required bandwidth as instead of transmitting a large number of local messages through several round trips, a mobile agent is transmitted to aggregate and report ambient data.

9. Load balancing can be improved in MA data aggregation as communication workloads (MAs) can be symmetrically scattered in comparison to client/server in which the workloads (data packets) focus on the communication links between particular nodes mainly source nodes and intermediate aggregators.

MA migration itinerary planning is a challenging issue in MA data aggregation. It is clearly related to the Travelling Salesman Problem (TSP) in which optimal itineraries are established for salesmen to follow. Although solving TSP (and similarly MA itinerary planning) is practical when the number of nodes to visit (i.e cities) is small, the problem is in general NP-complete. However, there are three key differences between TSP and MA itinerary planning: (1) TSP needs to visits all the nodes (i.e cities), whereas MA itinerary planning only visits the nodes which are desirable for the consumer. (2) there is a single salesman which travels through in traditional TSP, whereas MA itinerary planning focuses on routing multiple MAs throughout the network. (3) TSP assumes global knowledge, whereas MA itinerary planning in WSN does not.

MA migration itinerary planning needs to consider three issues to discover/establish optimal routes [Mpitziopoulos et al., 2009]: (1) minimising journey delay: it has the potential to enhance freshness of collected data that results in enhancing the accuracy of data analysis. (2) reducing network resource consumption (mainly energy): MAs need to move through short, low cost and efficient energy use routes as WSNs are highly resource constraints. (3) maximising the number of collected data samples: capturing greater number of data samples for data aggregation enhances data collection accuracy and robustness.

MA itineraries are usually designed in two fundamental schemes depending on where the routing decisions are made: proactive and reactive [Wu et al., 2004]. MAs utilise the itinerary that is created by sink or data consumer in advance of migration in the former, whereas MAs are routed on-the-fly according to the acquired routing informations that are dynamically collected in the latter. However, the MAs may use hybrid routing in which proactive and reactive itinerary planning are combined.

2.5.2.3 Proactive MA Itinerary Planning for Data Aggregation

Proactive MA data aggregation utilises a pre-defined list of nodes to sequentially visit them for data collection. The MAs use proactive itineraries to migrate as it has the potential to reduce routing overhead, energy consumption and data collection delay [Chen et al., 2007]. As the migration paths are defined (by sink and/or data consumer) priori to migration, the overhead of route discovery and establishment can be reduced. In other words, the cost of MA routing is reduced as MAs do not need to compute the routes on-the-fly. Moreover, the MAs return the aggregated results to the sink with the minimum possible delay because the sink will provide them the shortest itineraries to move. Proactive MA routing can be used in laboratory experimental situations because of its relative ease of installation. However, it is not applicable on most WSN data aggregation applications as the MA dispatcher (sink) should be aware of source nodes address, location, source events and the possible shortest paths between them. Sensor nodes may be randomly distributed in harsh and/or out of reach area like the ocean. Hence, collecting required routing information to design proactive itineraries would be difficult, expensive and/or impossible.

Mobile Agent Based Wireless Sensor Network (MAWSN) [Chen et al., 2006] proactively allocates a migration itinerary to a single MA to follow. The MA utilises a pre-defined routing which consists of three parts: (1) the first node id/location that the journey is started from, (2) intermediate nodes that need to be visited during the MA migration, (3) last node id/location that the journey should be ended in. According to the itinerary, the MA is moved from the sink to the first node through a proactive path to start the journey. Then, the MA selects the source nodes one by one from the itinerary list to visit. The nodes are ranked according to their distance to the sink in advance of the MA migration. Hence, the MA moves to the node which have closer distance to the sink at each step. The procedure is performed until all the source nodes in the list are visited by the MA. At the end, the MA moves to the last node to return the results through a reserved path to the sink.

Mobile Agent Distributed Sensor Network (MADSN) [Qi et al., 2003] moves a MA using a proactive routing map to collect and aggregate data samples similar to MASWN. The difference of MAWSN and MADSN is that, the latter utilises MRI (Multi Resolution data Integration) [Qi et al., 2001] technique to reduce data redundancy. Using this technique, source nodes estimate their data redundancy (using an overlap function) and then avoid to report if they are redundant. In other words, MADSN avoids visiting the source nodes that have redundant data. Hence, it reduces data collection overhead as compared to MAWSN. MADSN is suitable for clustered network in which the cluster heads perform MRI to discard redundant data.

Mobile Agent Directed Diffusion (MADD) [Chen et al., 2007] moves a single MA throughput the network in a hybrid manner for data aggregation. The MA itinerary is allocated to the MA similar to MASWN. The MA reactively migrates between first and last nodes that are proactively selected by the sink. It means that, the MA is moved to the first node by sink and then dynamically selects the closest node until it is reached by the last node. At the end, the MA returns to the sink from the last node through a reserved path. There are differences between MADD and MASWN: (1) MADD reactively visits the nodes between the first and last visiting node. It dynamically selects the closer source node to its current location to move in each migration. (2) MADD selects the farthest source node from sink to start the MA journey. The reason is that the author believes moving empty/lightweight MAs through longer routes in early migrations reduces network resource consumption. In fact, starting the migration from farther nodes using a MA in which data part is empty reduces communication overhead (size × communication distance) that has a high impact on energy consumption. (3) MADD discards the method part when the last source node is visited. It results in reducing transmitted network traffic as the MA does not transmit the method part to the sink.

2.5.2.4 Reactive MA Itinerary Planning for Data Aggregation

Reactive MA data aggregation derives MAs to visit the source nodes through paths that are dynamically formed [Wu et al., 2004]. The routes are established using routing information that are collected during data aggregation procedure at each node. For example, MAs select the closest node to migrate in the next using Received Signal Strength Indicator (RSSI) [Xu et al., 2010] technique which estimates the distance between the wireless nodes. Reactive MA routing has the potential to resolve existing drawbacks of proactive routing as it is flexible and scalable to deal with network topology and density changes. However, data collection delay is increased as MAs need to discover and establish migration routes on-the-fly at each node.

[Qi and Wang, 2001] proposes two algorithms named GCF (Global Centre First) and LCF (Local Closest First) to move a single MA into the event region(s). GCF moves a single MA to visit the source node which is closest to the centre of event region through shortest paths, whereas an MA is moved in LCF to the closest source node from the current location. They are comparatively simple to implement and have low computational complexity to route MA. However, data collection/aggregation cost and delay is increased when network size and/or density rises. The single MA needs to travel through long paths to visit the source nodes in large and/or dense networks [Mpitziopoulos et al., 2009]. Moreover, the performance of LCF and GCF highly depends on the current location of MA and/or event sources. For example, the MA would be able to visit the source nodes in GCF if the centre of event region is known by the sink. Although it is not critical when the source nodes are distributed in ER model (centralised), reporting the centre of RS (random distributed) event regions to sink/MA is expensive for WSNs especially when the network deployed is large and dense.

[Chen et al., 2009b] proposes two algorithms named IEMF (Itinerary Energy Minimum for First-source-selection) and IEMA (Itinerary Energy Minimum Algorithm) in which MAs are forwarded through minimum cost paths to collect data. Similar to LCF, the objective of IEMF is to reduce MA migration cost by selecting the minimum cost (consumed energy) link among all available ones to forward MA. IEMF allocates a estimated cost value to each route that is established to an event region. According to the cost value, it selects the closest node that resides on minimum cost link to migrate. The difference of LCF and IEMF is that LCF selects the closest node to current location of MA, whereas IEMF considers the estimated cost value on each tie to select the closest node to migrate. IEMA extends IEMF by selecting the next visiting source nodes in an iterative manner. Each available route to source regions is allocated by a cost value that is updated iteratively when the cost value of a node is measured. Indeed, IEMA considers a number of available links to event regions in an iterative manner to find out the route in which MA migration cost is minimised. As a result, it can be argued that LCF and IEMF are IEMA with zero and one iteration.

The Near-Optimal Itinerary Design algorithm (NOID) [Gavalas et al., 2010] utilises multiple MAs which independently travel throughout the network to collect and aggregate data samples. It increases the parallelism degree of data aggregation routing and consequently reducing delay as a number of MAs move throughout the network in parallel to capture and aggregate data samples. The MA migrations are started from the sink through the routes that are separately established for each event region. The routes are extended using a greedy algorithm to minimise a cost function that considers hop count and residual level of energy at nodes. NOID allocates a cost value to each link. It allows the MAs to select the closest node residing on the minimum cost link to move. In other words, the MAs move through links which minimise journey hop count (minimum delay) and have sufficient energy to guide them to source regions. NOID also considers the amount of collected data at each nods to control MAs size. As MAs become bigger/heavier when a number of sensor nodes are continuously visited (with no break), forwarding MAs without considering their size will increase the transmitted network traffic that results in increasing network resource consumption. For this reason, NOID monitors the MAs size at each node to avoid forwarding heavy MAs. It stops the migration and return the MA to the sink if its data parts become full and/or heavy. However, MA migrations to overleaped areas and capturing redundant data samples are the drawbacks of NOID. The MAs only consider the address of nodes instead of their available data during data aggregation journeys. In consequence, they visit overleaped area and capture redundant data. Besides, the complexity and/or overhead of managing multiple MAs increases especially when the number of event regions (RS model) in dense and large WSNs increases.

Genetic Algorithm based Multiple mobile agents Itinerary Planning (GA-MIP) [Cai et al., 2010] utilises Genetic Algorithm (GA) to compute routes for multiple MAs. This algorithm provides a random route map as initial gene that is shared between all MAs to avoid MAs collision and/or overlapped migrations. The route map has two vectors:

sequence and group. The former keeps the list of source nodes that need to be visited, whereas the latter indicates the number of source nodes that should be visited by each MA. This means that each element of group vector specifies the number of nodes that are listed in sequence vector and need to be visited by a MA. The vectors are updated/trained using the GA algorithm. GA operators (crossover and mutations) increase the variety of other possible route maps (gens) which are generated in an evolutionary manner according to the nodes visiting status and/or network topology changes. At each iteration, the best fitted route maps are selected using the selection operator. This procedure is repeated until the most efficient itinerary is achieved for the MAs to follow. The drawback of GA-MIP is delay. MA itinerary planning delay increases as the MAs need to optimise an initial pre-defined vector during data aggregation procedure. For this reason, GA-MIP is not suitable for real-time or time-sensitive data aggregation applications.

Tree-Based Itinerary Design (TBID) [Konstantopoulos et al., 2010] proposes a data aggregation protocols in which the MAs move through a number of spanning trees (SPTs) to collect and aggregate data samples over a zone-based network. Each tree is rooted in the single-hop neighbourhood of sink and assigned by a MA for data aggregation. First, TBID forms a set of concentric zones around the sink. Radius of each zone is N ×(maximum radio range of node)/2 in which N is the zone number. Then, each node residing in the first zone (single-hop neighbours of sink) starts to establish a spanning tree with the source nodes. To form the tree, source nodes which reside in outer zones are incrementally interconnected to the inner ones using a greedy algorithm. The inter-zone links form the tree trunk, whereas the intra-zone links shape the tree branches. This procedure is repeated until source nodes in the last zone (most outer one) are reached. At the end, the MAs start their journeys from the roots to visit all source nodes that reside on the tree branches. Each MA sweep all connected nodes to the tree in each zone and then move to the next one to visit next source nodes. The MAs return through the same infrastructure to the sink to deliver aggregated results. The drawback of TBID is to establish SPTs in proactive manner. It increases the network resource consumption in the case of frequent network topology changes. Moreover, complexity and cost of data aggregation is increased when the algorithm is implemented in large WSN as a greater number of SPTs need to be established.

[Seng et al., 2003] proposes a reactive mobile agent data aggregation protocol that is particularly designed for object tracking. It uses two types of MAs: master and slave. A Master MA is responsible for detecting the event regions and reporting aggregated results to the sink, whereas a slave MA collects data samples within a region to report to its master. According to it, a master MA is forwarded from the sink to the closest source node (minimum hop count) of each event region. Each master MA detects the current status of the event (i.e object location and direction) using the collected information at the source node. Then, it forwards a set of slave MAs from the source node to collect interesting data samples in the vicinity. In object tracking application, for example, slave MAs move to the sensor nodes that are able to dominate the movement space of the object. They report collected data samples to master MA in two schemes: threshold-based (TB) and distance-based (DB). In the former, slave MAs report if the tracking object meets the required threshold such as speed and/or direction, whereas the latter let slave MAs to report data if tracking object move to a area which is farther than a specific distance. Slave MAs collect and aggregate data samples and then report the results to the master MA. Master MA collects and aggregates the information from its salve MAs and then return to the sink to report. The drawback of this algorithm is the performance cost. The routing performance cost increases when the event/objects are distributed according to RS model. In this case, a greater number of MAs should be forwarded into the network to collect data samples from each event source. It increases network resources consumption. Moreover, data collection accuracy reduces if the MAs move into overlapping area to collect and combine data samples that can be highly redundant. Agent-based Event driven Route Discovery Protocol (AERDP) [SETHI et al., 2011] (and [Wang et al., 2007b]) aims to resolve the drawbacks by establishing a hierarchical infrastructure for MAs to migrate. The network is partitioned into a set of hierarchical groups (clusters) that are not overlapped. The groups are interconnected through a spanning tree which is rooted in the sink. Similar to [Seng et al., 2003], each region is assigned to a master MA to aggregate and report data samples that are collected by salve MAs. The overhead/complexity of this algorithm highly depends on the size and distribution model of event regions. It increases when the event regions are broadly formed with numerous and/or random events.

The introduced mobile agent routing protocols are highlighted and compared in Table2.14.

2.5.3 Summary: WSN Network Routing

Routing is an essential element in WSN to forward measured data samples from source nodes to data consumer access point (sink). It focuses on interconnecting data regions to sink using either single or multi-hop links that are formed according to the network architecture, application and/or data consumer requirements.

Network topology change is a challenging issue which influences the performance of routing in infrastructure-less network such as ad-hoc and/or WSN. Node mobility and residual energy level are two key parameters that usually lead to the network topology changes in infrastructure-less networks. Node mobility is the reason of most topology changes in ad-hoc (particularly MANET) networks, whereas the available level of energy at nodes is the key parameter which causes network topology changes in WSNs. Residual level of energy affects sensor node availability, wireless signal quality and wireless communication range. Hence, it influences wireless communications and consequently routing performance.

Routing in WSN is different with ad-hoc networks due to five reasons:

1. lack of global addressing scheme
2. limited resources at the sensor nodes to process, transmit and storage data packets.
3. it is sensitive to energy consumption as sensor nodes are equipped by limited and non re-chargeable power resources.
4. it highly depends on the application as a WSN is deployed for a specific scenario based on the consumer requirements.
5. it focuses on forwarding data packets from source nodes to the sink in a convergence manner.

WSN routing protocols are classified according to a set of parameters that is rooted in network architecture, consumer requirements and/or routing operation. The classifications would highlight routing protocols similarity, difference and/or superiority. The key classification of WSN routing is based on network architecture (flat and hierarchical) in which routes are established according to the nodes role. WSN routing protocols also is categorised according to the techniques such as location and/or energy awareness that are used to route the network traffic. The techniques are used in a routing protocol based on the consumer requirements and node capabilities.

Data aggregation offers a set of benefits such as reducing network congestion and energy consumption in WSN routing by reducing size/number of transmissions. Data aggregation routing focuses on two schemes: client/server and mobile agent. In the former, routes are established between the source nodes and intermediate aggregators or sink according to the network architecture and/or application requirements. The paths guide data packets from the source regions to the sink through a set of intermediate nodes which perform in-network data aggregation. In the MA routing, a single or multiple MAs move through proactive or reactive paths to capture and aggregate data samples. It offers a set of benefits specifically reducing network traffic, enhancing adaptability and autonomous computation as compared to client/server model. However, itinerary planning to establish efficient and low-cost paths for MAs is a challenging issue in MA data aggregation routing. Table 2.15 summarises and compares the key features of data aggregation routing in both schemes.

Table 2.13: Flat vs. Hierarchical data aggregation routing

Features	**Flat Networks**	**Hierarchical Network**
Aggregator nodes	any node on the path	intermediate aggregators i.d leader/CH
Node failures	disconnect network	locally disconnect clustered or grouped area
Traffic congestion	High (packets are forwarded by any)	low (aggregators forward data packets)
Message collisions	High	low
deployment cost	low	high
Node heterogeneity	doesn't matter	stronger nodes are selected as aggregators
Delay	High (multi-hop paths from event regions to sink)	Low (early aggregation)

Table 2.14: Mobile agent routing protocols

Protocol	Mechanism	Key advantage	Key drawback	Number of MAs	Routing metric
LCF	to closest node	1- ease of implementation 2- low overhead	1- AC node visit 2- increased delay 3- location dependent	single	shortest link
GCF	to centre of event regions	1- ease of implementation 2- low overhead	1- finding event regions 2- increased delay 3- location dependent	single	shortest link
NOID	to closest node which have enough residual energy	reliable migrations via the nodes with enough energy	overlapped data collection	multiple (single-hop neighbours of sink)	shortest link with respect to residual energy
IEMF and IEMA	iteration to find the minimum cost link (consumed energy and distance)	reducing MA migration cost	increased delay	single	minimum cost link
GA-MIP	GA for optimising MA routes	flexible to find best fitted routes based on consumer requirements	increased delay of route planning	multiple	best fitted routes based on consumer requirements
TBID	MA migration through a tree	reducing random walks of MAs	1- high cost of updating tree broadcast 2-a number of trees for large network	multiple (single-hop neighbours of sink)	shortest links
[**Seng et al., 2003**]	master/slave MA technique	reducing MA migration length	1- overlapped data collection 2-high cost for RS model	multiple	shortest path
AERDP	master/slave MA technique	not overlapped migrations	high cost for RS model	multiple	shortest path

Table 2.15: Client/server vs. Mobile agent data aggregation routing

Features	**Client/Server**	**Mobile Agent**
Communication	uni/multi/broadcast (depending on architecture)	(parallel)Unicast
Parallel processing	Yes	Yes
Automaticity	No	Yes
Message Structure	Simple	Complex
Communication Overhead	Number of relay nodes (depending on architecture)	Number/length of MAs
Message Failures	depending on architecture, traffic and/or energy	depending on residual energy of nodes
Accuracy of data collection	depending on data msg. failures	depending on itinerary planning
Delay of data collection	depending on traffic and/or architecture	depending on itinerary and/or number of MAs
Key advantage	simplicity in deployment	1- reducing network traffic 2- adaptability
Key drawback	message collisions/lost	MA complexity/security

Chapter 3

Mobile Agent Data Aggregation Routing

This chapter focuses on data aggregation routing based on the Mobile Agent (MA) model. Mobile agent data aggregation routing forwards either a single or multiple MAs to the event regions to collect and aggregate the data. The MAs visit the sensor nodes to collect the measured data samples if they match the sink interests. The data samples are aggregated at the MAs using an aggregation function. The aggregated results are returned to the sink by the MAs. A hybrid mobile agent based routing protocol (called ZMA for Zone-based Mobile agent Aggregation) is proposed, described and evaluated in this chapter. Section 3.2 describes the ZMA data aggregation routing protocol, focussing on the key aspects of ZMA which address the existing drawbacks and enhance the performance of MA data aggregation routing. Section 3.3 explains the experiments which are designed to test and evaluate the performance of ZMA, which include varying node count, area size and the proportion of interesting data samples. Section 3.4 reports the results of the experiments according to five key metrics which are usually used to test the performance of data aggregation routing protocols namely, total consumed energy, total number of captured data samples (accuracy), average end-to-end delay, MA hop count and total transmitted traffic. Each parameter is measured and discussed to evaluate the performance of ZMA in comparison to two selected MA data aggregation routing protocols namely NOID [Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010]. Section 3.5 discusses and concludes the key points of the results to highlight advantages and disadvantages of the proposed protocol. Section 3.6 provides a summary of this chapter as a quick overview.

3.1 Mobile Agent Routing for Data Aggregation in WSN

The mobile agent (MA) data aggregation model of routing utilises MAs to capture and aggregate data samples from the source nodes which are scattered across the network. The MAs move to visit the source nodes through paths which are established either proactively or reactively [Wu et al., 2004]. Proactive MA itinerary planning reduces network resource consumption and MA migration delay as the paths are established in advance and the MAs do not need to consume network resources to compute the routes during the data aggregation. Reactive MA routing allows the MAs to find the required paths on-the-fly according to the acquired routing information during migration. The objective of reactive routing is to establish dynamic paths to visit the sensor nodes that have interesting data to collect. In the case of network topology changes, reactive MA itinerary planning is better suited compared to proactive [Chen et al., 2007]. However, MA migration delay and network resource consumption is increased in reactive routing as the MA migration routes are computed at each node during the data aggregation routing. Hybrid MA itinerary planning combines the key features of proactive and reactive MA routing to reduce the delay and network resource consumption of reactive routing and to enhance the capability of proactive itinerary planning to deal with WSN topology changes.

MA itinerary planning influences data collection delay, accuracy and energy consumption. Blind or random walks of MAs result in increased end-to-end delay, network resource consumption (mainly energy) and reduce the accuracy of the aggregated data. Besides, the MAs may miss visiting a number of nodes and move through long (in terms of hop count) and/or high cost (in terms of consumed energy) paths if they randomly move through non-optimal routes in the network. For this reason, intelligent MA itinerary planning protocols are desirable to move the MAs through low cost and low delay paths in which the desirable source nodes are visited. In other words, the MA routing protocols need to minimise the cost and delay of data aggregation while maximising its accuracy.

This chapter proposes a hybrid mobile agent routing protocol that dynamically establishes data centric paths to forward MAs into the data regions. The MAs are in charge of collecting and aggregating data samples and then returning the results to the sink. The protocol aims to improve the performance of MA data aggregation routing by resolving the existing drawbacks specifically reducing the MA itinerary cost and delay, enhancing the MA migration intelligence and increasing the data aggregation routing accuracy. The proposed algorithm decomposes the event source regions into a set of zones that are formed in a data centric (DC) manner. At each zone, a set of nodes (called ZMACs for Zone Mobile Agent Coordinators) are selected to start the MA journeys according to the type, availability and density of interesting data samples. The performance of ZMA is independent from network architecture, event source distribution model and/or data homogeneity/heterogeneity. Its key objective is to increase data aggregation accuracy (collecting as many as possible data samples) while reducing data collection delay and energy consumption. To evaluate the performance of ZMA, two protocols are selected from the MA data aggregation literature according to their similarities, maturities and popularities: NOID [Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010]. Similar to ZMA, both protocols utilise multiple MAs that migrate throughout the network to capture data samples. NOID reactively forwards the MAs through address-centric links over a flat network, whereas TBID moves the MAs through a tree infrastructure which is formed in data-centric manner.

3.2 The ZMA Protocol

The Zone-based Mobile Agent (ZMA) protocol routes the MAs over a zoned network to collect and aggregate data samples. As Figure 3-1 shows, ZMA establishes a set of concentric zones from the sink until all the nodes are covered. The zones are created to localise the routing communications and guide the MAs. In each zone, a set of data regions are formed according to the sink's expressed interests. The data regions interconnect the source nodes which reside at a same zone and have the same (interesting) data type to report. Each data region is managed by a node which called the Zone Mobile Agent Coordinator (ZMAC). The ZMACs are selected using a leader election algorithm that works according to a weighting function similar to [Vasudevan et al., 2003]. They are in charge of initiating the MA data aggregation journeys. The MAs select the routes at each node using a weighting function which allocates a weight value to each link according to the node's data type, connectivity degree, proximity to the event regions and residual energy level. The MAs collect and aggregate data samples from the desirable source nodes and then return the results to the sink. The sequence diagram of ZMA algorithm is shown in Figure 3-2.

3.2.1 Network Model

The network model consists of three components: sink node(s), sensor nodes and event sources. The sink node(s) is/are the data consumer end-point to monitor the network performance. In other words, data consumer queries and requirements are defined at the sink node(s). Hence, the MAs need properly to return the collected data and/or aggregated results to the sink for further processing and analysis. The sink node(s)

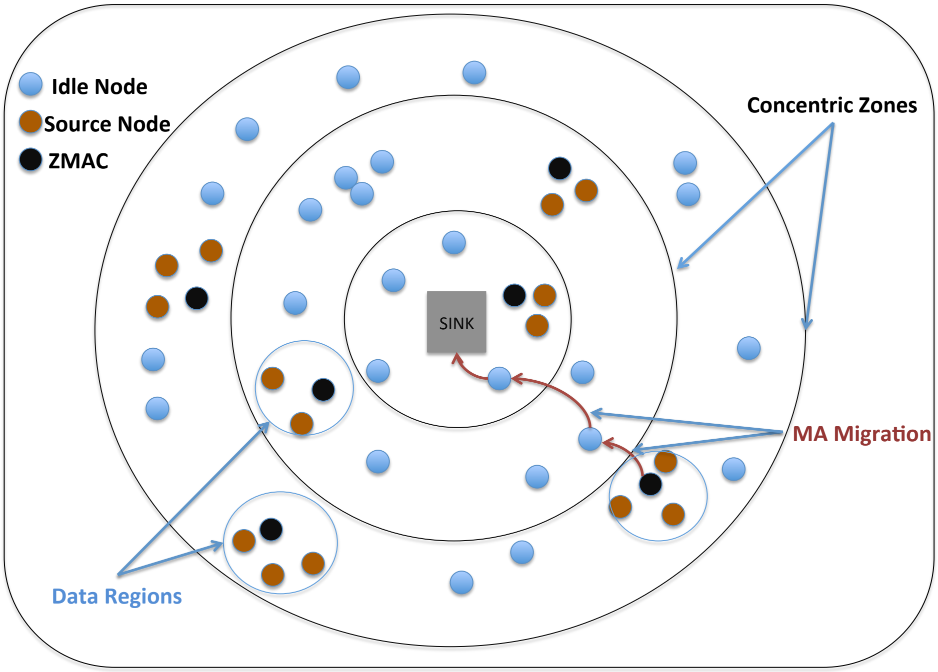


Figure 3-1: The architecture of ZMA

have sufficient resources for data storage, communication and/or computation. Such rich resources give the sink nodes exclusive abilities such as computing complex tasks, storing great amounts of information or direct communications to any node in the network.

The sensor nodes are responsible for measuring ambient quantities and/or forwarding the MAs. The nodes are static and have limited communication (radio range), computation and storage resources. No global addressing scheme is used in ZMA and the MAs migration paths are established in a data centric manner. Data centric routes are more desirable than address centric ones in WSN data aggregation routing protocols, as the data measured at nodes is more important than a node's address (IDs) [Krishnamachari et al., 2002]. The nodes can be homogenous or heterogeneous in terms of having variable level/unit of resources. The sensor nodes may be equipped with single or multiple sensing modules to sense a range of ambient events such as temperature, light and/or motion. It is assumed that nodes are synchronised to support message passing and wireless communications [Solis and Obraczka, 2004] The maximum communication radio range of each sensor node is R meters. This allows nodes to communicate within the region defined by a disc of radius R. It is assumed that the value of R is fixed/constant and known in advance of network deployment. The value of R influences network energy consumption and network connectivity. As energy con-

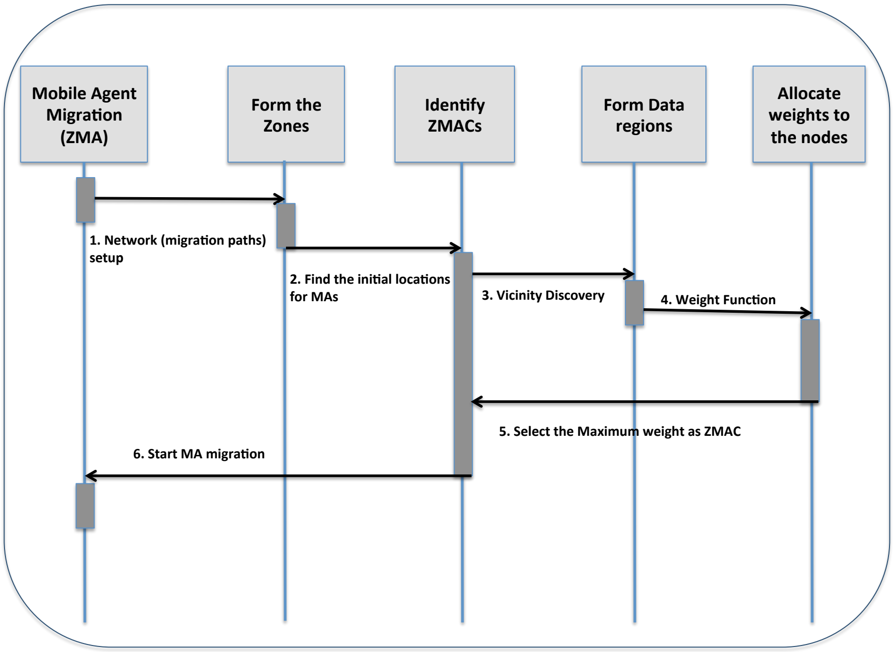


Figure 3-2: The sequence diagram of ZMA

sumption in wireless networks depends on the Euclidian distance between the sender and receiver nodes [Deng et al., 2007], a sender needs to consume a greater amount of energy when its R is increased. On the other hand, the sender node coverage and consequently network connectivity is reduced when R is small. For this reason, sensor nodes use the technique of wireless power adjustment in real applications [Song et al., 2009]. This technique allows the sender nodes to dynamically adjust their radio range (R) based on distance to receiver nodes for each transmission. This results in energy saving with respect to network connectivity.

Each sensor node plays at-least one of the following roles in the network: Idle, ZMAC, Source, TS (To the Sink). ZMACs are responsible for initiating the MAs migration in the network. They are selected in a distributed manner when the zones are formed. Source nodes are able to measure the desirable events which match the sink queries. TS (To the Sink) nodes know a reliable/short path to the sink from each zone. They are identified during the zone-forming phase. Idle nodes do not perform any particular task in the network. They are usually isolated unless they are asked by other nodes to perform back-up services during network deployment, data aggregation, mobile agent migration and/or network healing.

The event sources generate the environmental data in the region covered by the network. They may be either static or mobile. The sources are scattered across the network according to either Event-Radius (ER: 100% detection) and Random-Source

(RS: random detection) models. The source nodes measure similar data with different values depend on their proximities to the events. For examples, thermal sensors which are closer to a fire event report a greater temperature value as compared to other ones which are far away.

3.2.2 Forming the Zones

ZMA partitions the network into a set of concentric zones around the sink. It is started from the sink until all nodes are allocated a zone number. The zones are constructed for three reasons: (1) limiting the routing communications (i.e control packets) into the bounded regions (zones) to reduce overhearing and network resource consumption, (2) localising the MA migrations into the limited areas (zones), (3) guiding the MAs to return to the sink by moving from outer to inner zones. This allows the MAs to move in limited (DC) area instead of blind/random walk and/or heuristic migrations.

Hello messages ($Hello_{v1}$) are used in ZMA to form the zones. The hello messages are small/lightweight packets that are popularly used in ad-hoc networks (and similarly in WSNs) for vicinity discovery and/or determining the wireless link status (route maintenance) [Broch et al., 1998]. According to [Chakeres and Belding-Royer, 2002], the size of hello messages in IEEE 802.11 is between 20 to 512 bytes which is smaller than data packets that can be between 512 bytes to 10 Kbytes. Owing to this, and due to the fact that network resource consumption (especially energy and bandwidth) highly depends on network message size, utilising the hello messages in WSN is not expensive.

The zone forming phase starts when the sink broadcasts a $Hello_{v1}$ message (version one). The structure of ZMA hello messages is similar to WSN routing protocols such as directed diffusion [Intanagonwiwat et al., 2000]. As figure 3-3 shows, the $Hello_{v1}$ message has an additional field in its header which is the Zone Number (Z_{Nb}). Similar to TBID [Konstantopoulos et al., 2010], the messages are broadcast within a $\frac{R}{2}$ radio range to form the zones. The messages form a set of concentric $\frac{R}{2}$ width zones around the sink. This zone size is intended[1] to interconnect the nodes (with maximum radio range R) at zone (i) with at least one node in an outer ($i+1$) and an inner ($i-1$) zones. The value of Z_{Nb} is initially zero at the sink. Each node receiving the $Hello_{v1}$, increases Z_{Nb} by one and then updates the message with the new Z_{Nb} value for the next hop sensor nodes. A node updates its zone number according to the minimum received Z_{Nb} value. The minimum Z_{Nb} value indicates a minimum hop count path to the sink.

Each sensor node finds its TS node (To the Sink) during the zone forming phase.

---

[1] $\frac{R}{2}$ seems the obvious value. We have not experimented with changing this, though that would be a valid piece of further research.

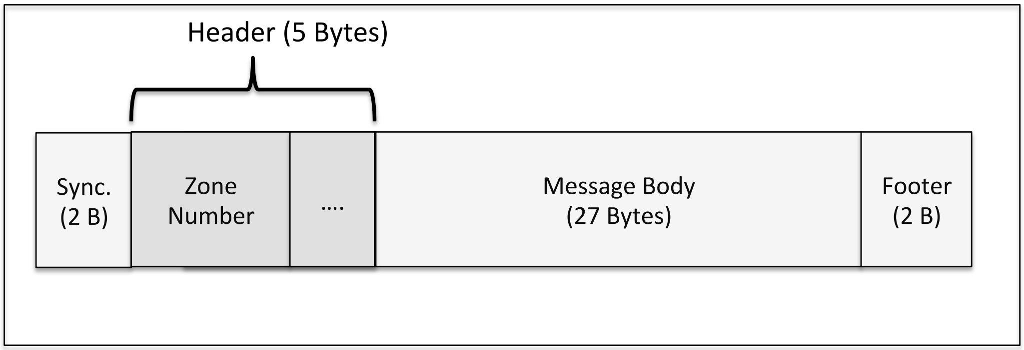


Figure 3-3: The structure of hello messages ($Hello_{v1}$)

TS nodes are responsible for providing reliable backward paths to the sink. A sender node of $Hello_{v1}$ with minimum Z_{Nb} plays the TS role for the receiver node at next (outer) zone. For this reason, each node at zone *i+1* keeps the id of the last sender node in zone i as its TS when its zone number is updated. The nodes also record a set of BackUp TS (BUTS) nodes if they receive multiple zone number from their single-hop neighbours at inner zones. BUTS nodes are used when the TS node fails or is not available. The $Hello_{v1}$ are re-broadcast within $\frac{R}{2}$ radio range until all nodes receive a zone number.

Message conflict is the drawback of utilising hello messages in WSNs. The message conflict is increased if the hello messages are frequently and/or simultaneously used by a great number of sensor nodes. The hello message lost ratio reduces in ZMA due to two reasons:

1. The vicinity discovery communications are limited to the zones. It reduces message conflicts as the number of sensor nodes which try to transmit the hello messages reduces.

2. ZMA allows the nodes to transmit the received hello messages over a uniform period time of (A, B). In other words, the received messages are re-broadcast after a short random time rather than immediately after they are received. This technique (similar to Minimum Cost Forwarding Algorithm [Ye et al., 2001]) has the potential to decrease the number of sensor nodes that simultaneously access wireless channels to broadcast the messages. This should result in reductions to the number of simultaneous/parallel transmissions and consequently the message conflict ratio. That is, the sensor nodes wait for a T_i which is calculated at each node using the equation 3.1 and then re-broadcast the hello messages for the next hop nodes. V_i is a random value which is selected from a uniform distribution of time values in the range (A, B). The random time range (A, B), for example

(10, 50) ms, can be set at each sensor node in advance of network deployment. A node (i), that receives the zone forming messages at received time (R_i), waits for a random period V_i and then re-broadcasts the message for next hop nodes at T_i. Therefore, the number of simultaneous messages at each zone may be decreased as they are uniformly re-broadcast over a time window.

$$T_i = R_i + V_i \tag{3.1}$$

Hello message failures during the zone forming phase influences the performance of ZMA as the nodes without the zone numbers cannot properly perform the MA itinerary planning. To resolve it, the nodes which miss or lose the hello messages $Hello_{v1}$ ask their neighbours to get the zone numbers. The nodes broadcast a zone enquiry message after a time period called Zone Time (ZT). The ZT depend on the maximum number of created zones in the network fields. According to figure 3-4, the maximum number of zones ($\frac{R}{2}$ width) which can be created in a N×N m^2 network is Max_Z. Hence, the maximum required time to finish the zone forming procedure can be calculated according to equation 3.2. $start_T$ is the network start time (wake-up time), Max (V_i) is the maximum value of the range (A, B), as explained earlier, and Com_T is the communication delay time that can be measured locally at each node [Ardakani et al., 2014]. After ZT, any node which has not already received a zone number broadcasts a zone enquiry message and then waits for Allowed Hello Loss [Perkins and Royer, 1999] to receive the reply. Allowed Hello Loss is a timing parameter which is used in AODV [Perkins and Royer, 1999] to control the transmissions and reduce message conflicts. It determines the maximum time that a node needs to wait before assuming a message failure. According to [Chakeres and Belding-Royer, 2002], two seconds is recommended for the Allowed Hello Loss. Each node that has a zone number and receives the enquiry message replies to the sender. The smallest zone number amongst all the received reply messages is selected, incremented by one and set as the zone number. The sender node of the respective message is recorded as a TS node.

$$ZT = start_T + Max_Z \times (Com_T + Max(V_i)) \tag{3.2}$$

3.2.3 Identifying the ZMAC Nodes

The ZMACs are responsible for initiating the MA migrations in each zone during data aggregation routing. They are selected using a weighting function similar to Common Election Algorithm (CEA) [Vasudevan et al., 2003]. The weighting function considers a set of features such as data type, connectivity degree, distance to the event regions

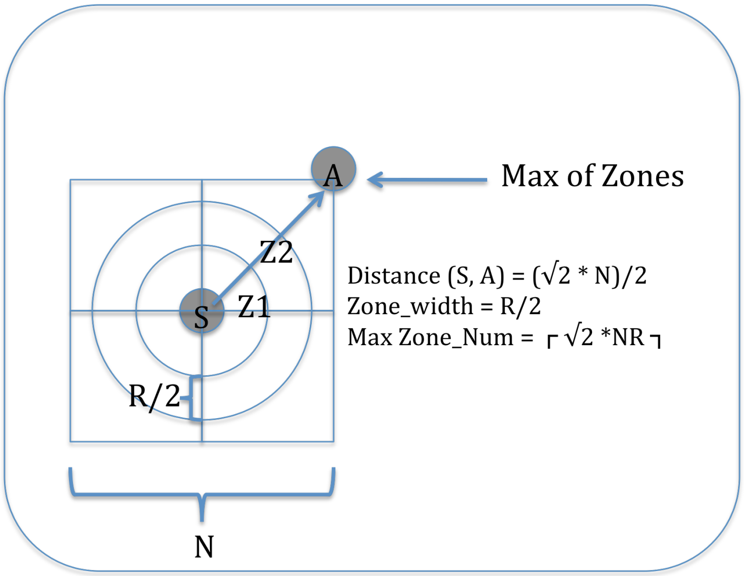


Figure 3-4: The maximum number of zones in a N×N network

and residual battery power to select the ZMACs. This means that a node is selected as ZMAC if it has enough energy level, maximum degree of connectivity to the desirable source nodes and minimum distance to the event source. That is, a set of data regions is formed for each zone with the source nodes which have same data types and zone numbers. The weighting function allocates a weight value based on the above features (energy, connectivity degree and distance to event source) to each node at each data region. The weight values reflect the chance of a node becoming a ZMAC. The node which has the greatest weight is selected at each data region as ZMAC. The procedure is explained in three phases namely vicinity discovery, weighting function and ZMAC selection in the following sections.

3.2.3.1 Vicinity Discovery

Vicinity discovery phase is performed after zone forming in ZMA. Each node may discover its local vicinity by finding available connections to any neighbour that has the same type data in its zone. They use hello messages $Hello_{v2}$ (version 2) for vicinity discovery. $Hello_{v2}$ has a similar structure to $Hello_{v1}$, however its header is slightly different. The message header has an additional field named data-type that is used to establish data-centric intra-zone links. When a $Hello_{v2}$ message is received, a data centric path is recorded from the receiver to the sender, if both nodes have the same zone number. This means that a $Hello_{v2}$ message is discarded if it is received from any node with different zone number.

```
Data: Routing Table (RT), Node ID (NID)
Node C:
if $Hello_{v2}$ message is received then
    if $Z_{Nb} = Zone(C)$ then
        Distance = RSSI (link);
        $RT \leftarrow (NID, Distance, D)$;
    else
        Discard ($Hello_{v2}$);
    end
end
```

Algorithm 1: Vicinity Discovery Algorithm

Each node measures the Received Signal Strength Indication (RSSI) [Xu et al., 2010] value upon the arrival of a $Hello_{v2}$ to estimate its distance to the sender node. A Line-Of-Sight (LOS) model [Uthansakul et al., 2005] is used by ZMA for wireless signal propagation when the nodes communicate with each other. This model assumes that there is no obstacle between the sender and receiver nodes and they communicate over a flat surface [Haslett, 2008]. Using this model, the receiver is able to estimate its distance to the sender using the RSSI technique when a wireless signal ($Hello_{v2}$ packet) is received. But in real life (as opposed to our simulation), obstacles affect the quality of the received signal and the receiver cannot estimate the distance if a Non-LOS (NLOS) model is used. In addition, it is assumed that there is no ambient noise affecting the wireless signals. The receiver is not able to measure RSSI and estimate the distance in case of environmental noise as it reduces the quality and reliability of wireless signals. According to equation 3.3, the power of the received (P_R) and transmitted (P_T) signals depends on the distance (d) between the sender and receiver and an environmental value (n). This means that a stronger signal is received from a closer sender. The formula is based on the reduction of the power of wireless signals due to fading effects over communication distances. RSSI is measured using the power of sent and received signals according to equation 3.4. A receiver node is able to estimate its shortest Euclidean path to a sender node, if the RSSI value is maximised. This means that RSSI value increases when the sender node is closer as the receiving signals have greater power. The RSSI technique is suitable for WSN as its implementation cost and complexity is low [Khan et al., 2012]. It has been used in a number of WSN routing protocols (i.e LEACH [Heinzelman et al., 2000])to measure distance between the nodes and/or establish minimum cost paths. According to algorithm 1, routing tables are updated with the received $Hello_{v2}$ messages. The routing tables allow nodes to find their neighbour's Id, available data type and distance (RSSI). After updating the routing tables, each node is able to calculate the number of its neighbours that have a particular data type to report. The routing tables are used in the next phases to form data centric regions and select the ZMACs to start the MA journeys.

$$P_R = (P_T) \times (1/d)^n \tag{3.3}$$

$$RSSI(dB) = 10 \log \frac{P_R}{P_T} \tag{3.4}$$

3.2.3.2 Weighting Function

ZMA utilises a weighting function (similar to NOID [Gavalas et al., 2010] and LEACH [Heinzelman et al., 2000]) to find the nodes that are eligible to become ZMACs for each zone. The weighting function returns a weight value for each node according to its connectivity degree, residual energy level and distance to the event sources. The nodes with higher weight value ($W_{(i,j)}$) have a greater chance to become a ZMAC. In other words, a node is selected as a ZMAC if it has the highest level of residual energy, greatest data centric connectivity degree and shortest distance to the event source.

The weighting function is calculated at each node according to equation 3.5. Each node calculates its own weight for each data type that is measured in its vicinity. The weight value of node (i) is correlated to the connectivity degree ($Count_{(i,j)}$), average distance to the event regions ($P_{(i,j)}$) and the proportion of residual level (E_C) to the initial amount (E_T) of energy. According to the function, the value is increased if the node resides closer to the centre of the event region and has a higher number of connections.

$$W_{(i,j)} = (Count_{(i,j)} \times P_{i,j}) \times \frac{E_C}{E_T} \tag{3.5}$$

To calculate the weight, $Count_{(i,j)}$ and $P_{(i,j)}$ are computed at each node. First, the collected information from vicinity discovery is classified at each node based on the measured data types to rank the connectivity degrees in a DC manner. Second, $Count_{(i,j)}$ and $P_{(i,j)}$ are calculated based on the classified DC links in two steps: (1) $Count_{(i,j)}$: is the total value of available links for data type j at node N_i. (2) $P_{(i,j)}$: is the average distance that shows the proximity of a receiver and sender node with respect to the type of data measured. It is used to establish short, low-energy links to the source nodes. That is, the Count values are ordered at each node according to the type of measured data. This means that the discovered neighbour source nodes are classified in a DC manner using the sum values. Then, each node calculates its average distance to the neighbour source nodes based on the measured data type. Similar to [Shen et al., 2005], ZMA utilises equation 3.6 to calculate the average distance to the neighbour source nodes. In the equation, $P_{(i,j)}$ represents the average RRSI value of the links which are established based on data type j at node N_i. For example, let us assume node A receives five and three $Hello_{v2}$ messages respectively from its neighbour nodes that report pressure and temperature. Hence, node A classifies its links into two categories (pressure and temperature): pressure ($Count_{(A,pressure)} = 5$) and temperature ($Count_{(A,temp)} = 3$). Node A calculates two average distances to pressure ($P_{(A,pressure)}$) and temperature ($P_{(A,temp)}$) regions using equation 3.6.

$$P_{(i,j)} = 10^{\left(\frac{\sum_{k=1}^{Count_{(i,j)}} RSSI(i,k)}{Count_{(i,j)}} \times \frac{-1}{10}\right)} \tag{3.6}$$

3.2.3.3 ZMAC Selection

The ZMACs are selected in a distributed manner using the weight values. The centralised leader election approaches need to collect the selection/weighting information at one or a set of particular node/s (i.e the sink) to detect, compute and allocate the roles. Although it is robust and has the potential to select and allocate the role in an optimal manner, it is expensive for WSNs. The centralised role allocators need to transmit a set of messages to collect the required information at role allocator node(s), inform the network nodes about their roles and/or update the new leaders/coordinators in case of network topology change. Hence, the communication overhead and consequently resource consumption is significantly increased if the network deployed is large and dense. Besides, the centralised pattern is not fault tolerant, as the performance of role selection depends on the availability of role allocator nodes (i.e the sink). This means that the role allocation procedure fails if the role allocator nodes fail. For this reason, ZMA does not use a centralised pattern to select ZMACs. It uses a distributed pattern in which the leaders/coordinators are locally selected by the nodes without communicating with any centralised node to detect and/or allocate the roles.

```
Data: Routing Table (RT), Node ID (NID)
RT Update at Node C:
if $Hello_{v3}$ message is received then
    if $Z_{Nb} = Zone(C)$ then
        $RT \leftarrow (NID, W(i,j), D)$;
    else
        Discard ($Hello_{v3}$);
    end
end
ZMAC Selection at Node C:
$Weight \leftarrow Max(RT.W)$;
if $Weight >= W(C)$ then
    SELECT Min(NID) from RT WHERE W = Weight As ZMAC;
else
    $ZMAC \leftarrow C$;
end
```

Algorithm 2: ZMAC Selection Algorithms

According to Algorithm 2, the nodes broadcast a hello message (version 3) called $Hello_{v3}$ whose header has two additional fields: D_i and $W_{(i,j)}$ to select ZMACs. The first is the sender data type and the second is/are the weight value(s) of data type(s) that is/are measured from the single-hop neighbourhood. For example, if it is assumed that node B measures pressure and its neighbours report both pressure and temperature, it broadcasts a $Hello_{v3}$ whose header consists of (D_B = pressure, $W_{(B,pressure)}$ = X, $W_{(B,temp)}$ = Y) after calculating the weight values X and Y using equation 3.5. A node may update its routing table when a $Hello_{v3}$ message is received. According to the example, node A updates its routing table by ($W_{(B,pressure)}$ = X, $W_{(B,temp)}$ = Y) and ($W_{(C,pressure)}$ = Z) when it receives the $Hello_{v3}$ messages from node B and C respectively. At the end, each node is able to find the greatest weight value for a received data type in its routing table. The node with greatest value is selected as a ZMAC for the respective data type. Otherwise, a node considers itself as the ZMAC for its zone if none of its neighbour reports a greater weight value. In the case of having the same weight values (based on a particular data type), the node with smaller Id is selected as the ZMAC. The ZMACs wait until they receive the sink queries to migrate the MAs for data aggregation.

The ZMACs are able to provide data centric and short links to the source nodes for the MAs to migrate. They also have sufficient energy to manage initial MA migrations. The ZMACs are selected using the weight values which are collected during the vicinity phase. The weight values are ranked at each node as a list which shows the neighbour's chance of becoming the ZMAC for particular data types in later rounds. The weight ranking is based on the routing information which are proactively collected at routing tables. It will be changed if the routing tables are updated due to topology changes (i.e node failures that will be discussed in page 97). Owing to this, the current ZMAC is able to find out the node that is eligible to become ZMAC (the next node in the ranked list) for the next duties. When the energy level of the current ZMAC falls below the threshold, it finds the next eligible ZMAC from its list to pass the role to continue the procedure. The current ZMAC firstly checks the availability of the candidate node and then sends a role exchange message if it is available and has enough energy to start the MA migrations. The new node becomes the data region ZMAC as soon as receiving the role exchange message from the last ZMAC without requiring any additional cost. However, the nodes may need to update their tables and select the new ZMAC (using Algorithm 2) in the case of topology change and/or falling the residual energy level below the required threshold to stay alive and/or maintain the connections. This issue will be discussed in the next section.

3.2.4 Mobile Agent Routing

ZMA assumes that the sensor nodes are initiated by MAs in advance of network deployment. The MAs become active to migrate and collect data using the aggregation codes when data collection requests from the sink are received by ZMACs. The data collection requests are propagated by the sink via indirect or direct communications:

1. The sink broadcasts the data collection hop-by-hop (similar to zone number propagation) until all ZMACs in the network receive. In this case, all the network nodes (namely source and/or ZMACs nodes) need to stay available to propagate the data collection requests from the inner to outer zones. This approach would increase the network resource consumption especially when the data collection requests are frequently propagated by the sink.

2. The sink directly sends its requests to a set of particular zones. This approach allows the source nodes to go to sleep as they are not required to receive and forward the data collection from the sink. This can reduce energy consumption, network congestion and traffic. In real applications, the sink usually has rich communication abilities to directly communicate to any node in the network. In other words, the sink adjusts its radio communication range to send a message

> to a particular part for network. For example, the sink sets its radio signal to cover circular disk of radius R. As the node communication is R and zone width is $\frac{R}{2}$, it covers the nodes which reside in first and second zones. Moreover, the sink would code the signal to cover the nodes which reside in particular zones. In this case, the nodes which reside in a particular zone, for example zone 2, would be able to receive the data collection requests if they are coded to be received by the nodes which have zone number two (Z_{Nb}= 2). As a result, the ZMACs stay on duty until they receive a request which matches their zone numbers.

The MA journeys start from ZMACs nodes in each zone to collect and aggregate data. In other words, ZMACs initiate the MA migrations upon receiving the queries according to data type that the sink is looking for. The sink may propagate variant queries for collecting different data types in the zones. For example, it is interested in collecting light information in zone (i), and temperature in zone (i+1). The ZMACs which match the sink queries update their MA codes according to the sink interests and then start to move the MAs using ZMA mobile agent routing algorithm which is explained in the next paragraphs.

The structure of MAs in ZMA consists of four components: identification, data space, code part and itinerary. The identification provides the identity information of the MA and its dispatcher, the data space stores the aggregated data, the code is the aggregation function and the itinerary provides the MA routing information. Although the MA identification, data space and code parts are effectively the same as other MA data aggregation protocols such as NOID [Gavalas et al., 2010] (explained in page 74) and TBID [Konstantopoulos et al., 2010] (introduced in page 75), the itinerary part of MAs in ZMA is different. It consists of four fields: next node ID, MP IDs, visited nodes list, non-visited nodes list (NV_List). According to figure 3-5, the next node id shows the address of the source node which the MA goes to next. MP IDs are the list of Meeting Point (MP) nodes referring to the nodes that should be visited in next migration rounds. The MP nodes have multiple links to the interesting source nodes. They are visited again during next MA migrations if any of their neighbours is missed out. The visited node list consists of the node addresses which are visited already, whereas the non-visited is the list of source nodes that are not visited yet and should be captured in the next migrations. Each non-visited list refers to a MP which has links to the non-visited nodes. Hence, each MA may return to the MP nodes (of the NV_List) which are awake and still have links to the non-visited nodes.

The MAs collect the first data sample from their respective ZMAC and then find the next node that has the greatest $W_{(i,j)}$ to visit next. The routing tables provide the required information for MA migrations. The chance of capturing data samples is

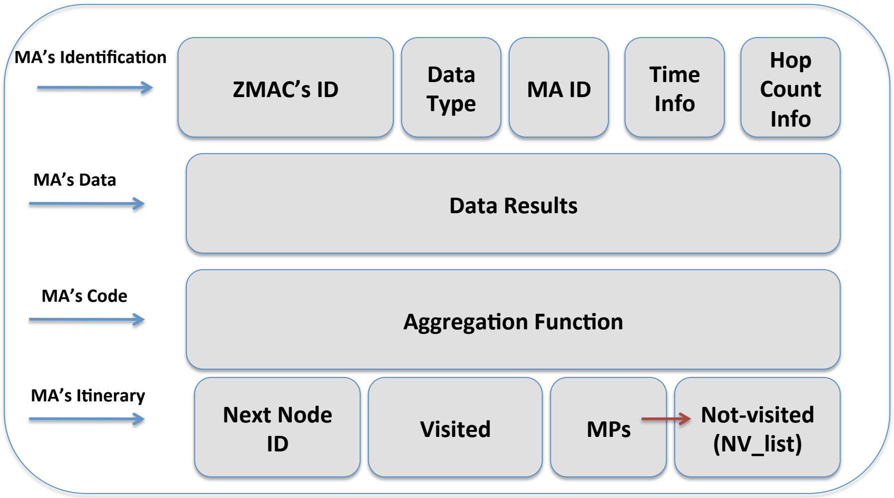


Figure 3-5: MA Structure in ZMA

increased in ZMA, because the MAs use the weight values to move to the nodes which have the greatest number of DC paths to the interesting data samples. This means that a MA selects the next hop node amongst all candidate nodes if that node has reliable paths to a greater number of desirable source nodes. According to figure 3-6, the MA considers the routing table at each node to find the next hop. In the simplest case, the MA finds just one node in the routing table that matches the sink query. The next node ID is set to the node id (NID) and then the MA moves to collect and aggregate the data. If multiple nodes are found in the routing table, the MA marks the current/host node as Meeting Point (MP) to return to later for further migrations if it is required. Then, it selects one node which has the greatest weight to migrate to. The remaining possible nodes/links are stored as a list (NV_List) of the MP in the MA itinerary part. According to figure 3-6, the MA migrate to the nodes and removes their Ids from the list one by one when they are being visited during the journey. The procedure is repeated until the MA visits a node that has no more links to the source nodes. In this case, the MA checks its NV list to find if there is any non-visited node. If there is, the MA returns to the MP of the list using its recorded journey to change the migration direction to visit the non-visited nodes. Otherwise, the MA prepares to return the aggregated result to the sink. In this case, the MA finds the TS address of the node to migrate to. TS nodes are recorded during the zone forming and are used to establish a reliable and shortest (minimum hop count) path to the sink. The MAs visit the TS nodes, one by one, on their return paths until are received at the sink.

In the case of node failures, ZMA performs a particular mechanism depending on the failing node role. It is assumed that the network topology (link/node failures) is

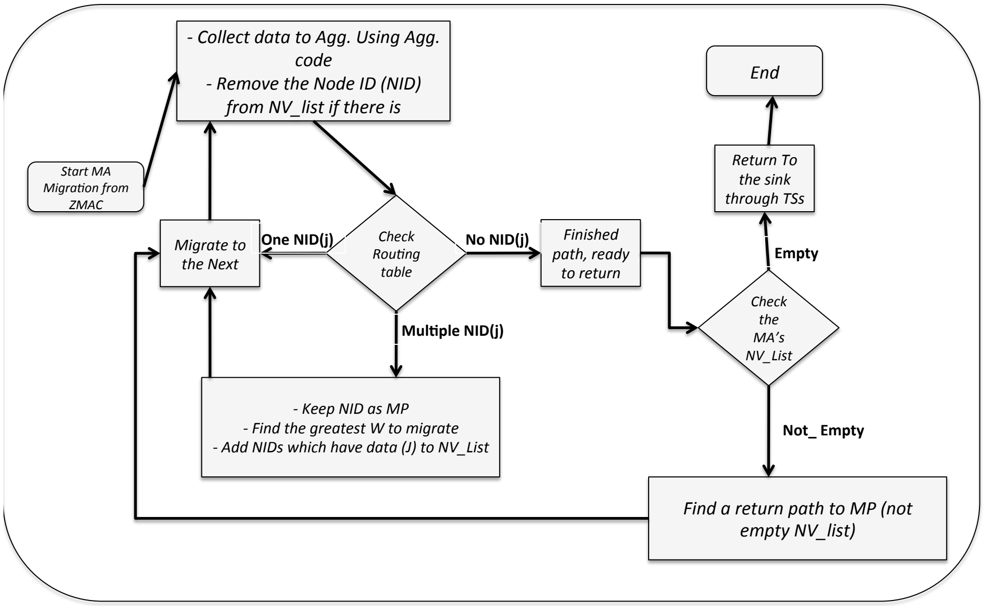


Figure 3-6: MA migration chart in ZMA

changed in ZMA if the available residual energy level falls below the required threshold to maintain the minimum connectivity between the nodes and/or keep the node alive. Unexpected node failures such as hardware damage and/or node capture attacks will be addressed in Chapter 6 as future work. If the node is a source node, it sets its weight value to zero and then broadcasts a message to inform its neighbours. The message lets the neighbours know that the node in their vicinity failed and there is no more link/data to follow. Then, the neighbours update their routing table with the zero value for the entity and hence avoid routing MAs to that node any more. If the node is an MP, it needs to find another node in its vicinity that has the ability to minimise the disconnections caused by the failures. This means that the new MP should have the ability to cover the maximum possible number of source nodes in the vicinity of the failure. In this case, the failing MP broadcasts a message called $Fail_{(MP)}$ to inform its neighbours of the failure. The message is attached also by a list of the source nodes that need to be covered for MA visits in next migrations. Each node which receives the message replies back if it has available links to any of the nodes. The reply message is attached by the list of requested ties (if they are available) and the sender node weight value. The failing MP selects the node with greatest weight as it has the ability to cover broadest area (greatest number of non visited nodes) amongst all nodes that received the failures messages. The new MP begins to play the role as soon as it receives the confirmation from the failing MP node. If a ZMAC node is failing, it broadcasts a

$Fail_{(ZMAC)}$ message. Each node which receives the message updates the weight value of the failing ZMAC to zero. Then, the nodes perform algorithm 2 to select the new ZMAC as it is explained earlier in section 3.2.3.3.

3.3 Experimental Plan

OMNET++ [OMNET++, 2012] is an open-source, component-based and discrete event simulation that is used to simulate ZMA routing protocols in order to test and evaluate its performance and efficiency. This simulator has a modelling framework called MiXiM [Viklund, 2013] for mobile and/or fixed wireless networks such as wireless sensor networks, MANET and VANET. It offers detailed models of radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols (i.e B-Mac)[Kpke et al., 2008].

The experiments in this chapter focuses on MA data aggregation routing and are designed to test and evaluate the performance of ZMA in terms of advantages and drawbacks. The results are compared to two existing works – NOID[Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010] – that also utilise multiple mobile agents for data aggregation. The experiments measure a set of metrics which are broadly used in the literature to evaluate the performance of MA data aggregation routing protocols [Wu et al., 2004], [Chen et al., 2009a], [Boulis et al., 2003]: total consumed energy, total number of captured data samples (accuracy), average end-to-end delay, MA hop count and total transmitted traffic[2]. Energy consumption needs to be measured as energy is a vital parameter in WSNs. Accuracy and delay should be considered in data aggregation routing because they are based in data consumer requirements such as maximisation of accuracy or minimisation of delay. In addition, MA hop count and transmitted network traffic are measured as they have a high impact on energy consumption, accuracy and/or end-to-end delay. This means that investigation of path hop count and network traffic should highlight the efficiency of routing algorithms to save the energy, reduce delay and enhance accuracy.

Figure 3-7 depicts the experiment parameter sweep plans. Accordingly, the experiment scenarios are designed over three levels of parameters: area size, node count and data density. The parameters let us observe each routing protocol's behaviour, scalability and performance according to varying area size (small, medium and large), node count (sparse or dense), and data density (25 to 100 percent). The experiment parameters are explained as below:

---

[2] The term total transmitted traffic is used throughout this dissertation to refer to the total sent and received network traffic.

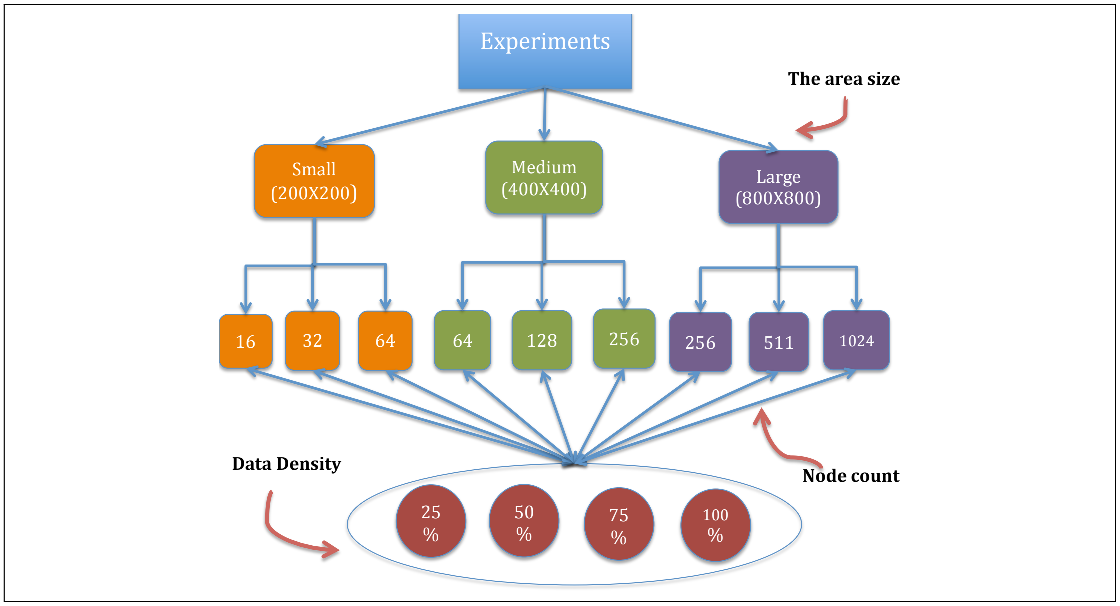


Figure 3-7: Experiment parameter sweep plan

1. Area size: this affects the wireless communication type (single or multi-hop) and consequently the performance of routing. The sensor nodes are able to communicate in single-hop in small networks, whereas they need multiple hops when the network area is increased.

2. Node count (node density): varies the number of nodes (one of which is the sink) in the network to test the protocol's scalability.

3. Data density: is the number of desirable source nodes in the network. This parameter allows us to observe the ability of MA routing protocol to find and capture interesting data samples when the proportion of desirable source nodes is varied in the network.

3.3.1 Simulation Setup

All the experiments are implemented using MiXiM [Viklund, 2013] that is used to simulate wireless sensor network. The key assumptions/issues of the experiment setup are outlined as below:

1. The experimental scenarios are designed according to WSN environmental monitoring applications as set out in Chapter 1 (Section 1.2). The sensor nodes report the measured data samples from a two-dimensional (2D) area using MAs to the sink for further processing and analysis.

2. The network is deployed with a single sink residing in centre of the field and a number of static sensor nodes that are randomly scattered. For all the experiments, it is assumed that the sink has a sensor attached which is able to measure the ambient events. This sensor is used to give the consumer a high level view to find out if the desirable data is available in the area to collect. This has the potential to enhance the automaticity of the system. To clarify, let us assume a system which is setup to detect and measure air pollution. The system will automatically activate the MAs to move for data aggregation when the sink's sensor detects the level of pollution is above a certain threshold in military operations if chemical weapons are used. Conversely, the system needs an online administrator to keep lookout and run the data aggregation procedure (MA activation) if the sink has no such attached sensor.

3. The sensor nodes are modelled as a set of static wireless nodes that are randomly distributed in the area. The power is supplied by limited lifetime batteries that are 99999 mAh and 3.3V. The node radio range is 75 metres (similar to TelosB motes in outdoor communications [Crossbow, 2014]) and communications are symmetric [Tse and Viswanath, 2005]. The nodes communicate over the B-MAC protocol (Berkeley Media Access Control for Low-Power Sensor Networks) [Kabara and Calle, 2012].

4. Data samples are allocated to a proportion of the sensor nodes according to a data density ratio which is explained in the previous section. The sensor nodes, which have the data samples are called source nodes. Hence, data samples are collected and reported from the source nodes which are randomly placed in the network, to the sink.

5. All the experiments are set up based on a Line-Of-Sight (LOS) model for wireless signal propagation. As it was explained on page 91, this models assumes that there is no wireless interference caused by physical objects (obstacles) and environmental factors (e.g weather conditions). The obstacles such as tree, wall and building will lead to weakening or even blocking of wireless signals depending on the density of materials used in them [Baguena et al., 2012]. Environmental conditions will influence the wireless signal integrity [NADEEM et al., 2010]. For example, lightning will cause electrical interference, while fog weakens the signals as they pass through.

6. All the experiments are run over a period of 3600 seconds, to have sufficient time to complete the set-up process and perform a reasonable number of data

aggregation routing attempts. The network is set up in each experiments within the first 50s.

7. All the experiments are run 300 times. Statistical power analysis technique [Walker, 2013] was used to determine the necessary repetitions. This technique is used in experimental design to determine the sample size required to detect the effect of a given size based on a degree of confidence. In other words, this technique calculates the number of repetitions (sample size) using the population standard deviation (based on a subset of experiments/samples) and according to a confidence degree. We run the experiments over small networks for 50 times (sample size) and measured then the standard deviation of total consumed energy for each protocol. The greatest standard deviation is used to calculate the minimum number of required repetitions as it shows the widest confidence interval. This means that the protocols (e.g ZMA) which have a smaller standard deviation need a fewer number of repetitions to achieve the confidence degree as compared to the protocols (e.g TBID) which have greater ones. According to [Serqeant, 2014], we calculated the minimum number of required repetitions to achieve 90% confidence using the assumed population standard deviation (the greatest standard deviation). Hence, we run the experiments for 300 times which is greater than the minimum required repetitions for all the protocols.

8. According to routing classification in Chapter 2, the benchmark protocols are selected from both flat and hierarchical categories to test and evaluate the performance of ZMA. NOID[Gavalas et al., 2010] is a flat MA routing protocols which reactively forwards the MAs, whereas TBID [Konstantopoulos et al., 2010] establishes a proactive hierarchical infrastructure for MA routing.

The experimental scenarios are designed to test and evaluate the routing protocols at three levels: area size, node count and data sample proportion. In the first level, the network is deployed in three different area sizes: small (200×200 m^2), medium (400×400 m^2) and large (800×800 m^2). This allows observation of the protocol's behaviour and performance in big and small networks. In small ones the communications between the nodes are mostly single-hop, whereas they may become multi-hop as the area is increased and the network is deployed a larger area. In other words, most nodes (with 75 meter radio range) in the small network are able to communicate in single-hop with the sink, whereas they may need to establish multi hop links in larger networks.

To test protocol scalability, a varying node count is considered for each area size. Deploying networks with a variable node count (node density) lets us observe the protocol's behaviour, scalability and performance in sparse and dense networks as the node count increases. A minimum required number of nodes ($Count_N$) to deploy a wireless network is calculated based on equation[3] 3.7[Zaidi et al., 2009]. N is the number of nodes, R is the maximum radio range, O is the overlapping area between nodes radio range, and M and K are the dimensions of the network field. Accordingly, each network is set up with a minimum number of nodes that is required to provide a connected network in the area. Then, the node count is increased with respect to the density which is calculated using the equation 3.8 [Youssef et al., 2009]. This means that first the protocols are tested over a small network (200×200) that is deployed with node count of 16, 32 and 64 (one of which is sink). The same experiments are then performed in medium (400×400) and large (800×800) areas with – in order to produce similar levels of node density – 64, 128, 256 and 256, 512, 1024 nodes, respectively.

$$Count_N = \left\lceil \frac{0.5 \times (M \times K)}{(R - (0.5 \times O))^2} \right\rceil \tag{3.7}$$

At the third level, each experiment features one of four proportions of nodes which have interesting data samples (data density) to report. Hence, each node count in each area size is allocated with four different data densities (25%, 50%, 75% and 100%) that need to be collected/reported. Appendix B.1 shows the simulator experiment setup file for ZMA for a network with 256 nodes in a large area (800×800) when 100% of node have data.

$$D = \frac{N}{M \times K} \tag{3.8}$$

3.3.2 Isolated Nodes

Random node distribution can lead to arise isolated nodes. A sensor node is isolated if it is not able to communicate with the sink (or any other node) either through single or multi-hop links. Isolated nodes influence the performance of routing protocols in terms of coverage and network connectivity. The network coverage and connectivity is reduced as isolated nodes are not able to report sensory data from their area to the sink or forward routing messages to the any other connected nodes. Hence, the area under their coverage is totally disconnected to the sink and becomes out of reach for the network consumer. In addition, isolated nodes waste network resources – mainly energy – with data sample collection and computation of routes that are not used. Owing to

---

[3] To find out the number of uniformly positioned (grid) nodes to fully cover a 2D area, factor 0.3125 should change to 0.5 in this equation. The original equation (with factor 0.3125) does not consider the uncovered area which is formed among each four sensor nodes that are placed in a 2×2 grid. Owing to this, factor 0.5 should be used as one node is required to fill the uncovered area for each four nodes.

this, isolated nodes should be considered during network deployment to avoid network disconnection or wasting resources.

The minimum required number of sensor nodes to cover a $M \times K$ meter area is calculated (i.e 15 nodes for the sparsest network in the smallest area) and used then to set up the experiments using equation 3.7. However, this equation measures the minimum required number for a uniformly-placed (grid) network rather than a random one. Therefore, we use equation 3.9 [Bettstetter, 2002] to calculate the probability of having isolated sensor nodes in our experiments as the nodes (except the sink which always resides in the middle) are randomly placed. R is radio range, P is the probability of no isolated nodes in a network, N is the number of (random-placed) nodes and D is network density(calculated using equation 3.8). This equation shows satisfactory probabilities (i.e 98.6% for 16 nodes in $200 \times 200m^2$ area) to deploy a network with no isolated nodes according to our experiment setup.

$$R \geq \sqrt{\frac{-\ln(1 - P^{\frac{1}{N}})}{D \times \pi}} \tag{3.9}$$

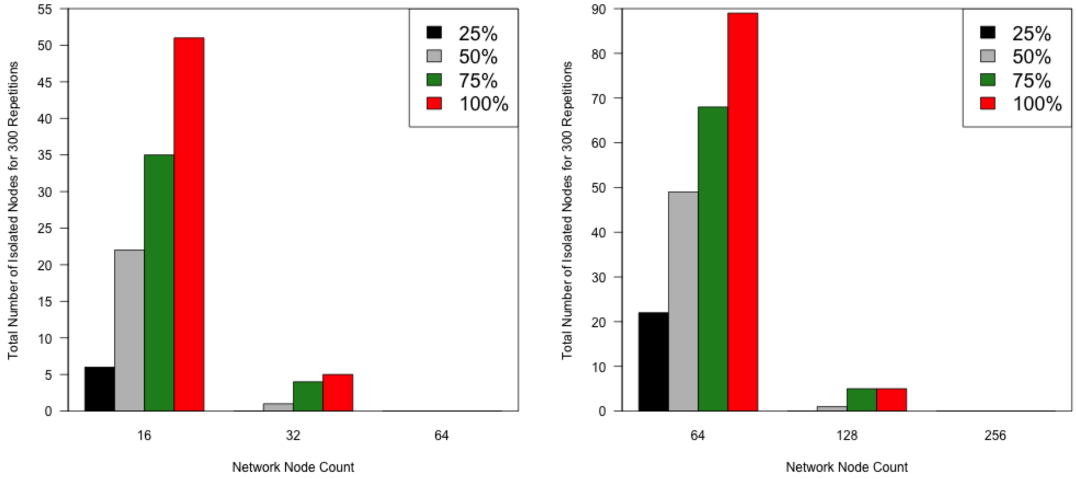
According to [Bettstetter, 2002], the probability of a network deployed with no isolated node (there is at least one link between any pair of nodes) is calculated using equation 3.9. However, it does not consider a group of nodes which are able to communicate with each other but placed out of the sink's reach. None of the grouped nodes is isolated, whereas the whole group cannot reach the sink to report data. For this reason, we measured the number of isolated nodes which cannot hear the sink in our experiments. Figure 3-8 shows the total number of isolated nodes for 300 repetitions per each node count and network size. As it is observed from the figure, the number of isolated node increases when area size increases. However, isolated node count decreases when node count (network density) increases. For example, there is no isolated node when the deployed network is dense.

3.3.3 Performance Parameters

Various quantitative metrics are typically used to analyse the performance of the routing protocols. The parameters chosen to be measured in this chapter are [Wu et al., 2004], [Chen et al., 2009a]: total consumed energy, total number of captured data samples (accuracy), average end-to-end delay, MA hop count, and total transmitted traffic. They are briefly explained in turn as below:

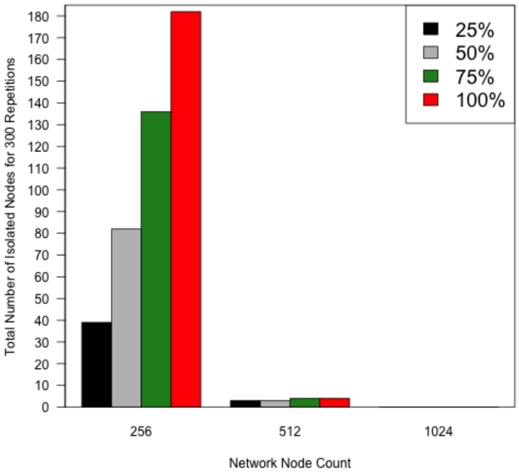
1. **Total consumed energy:** represents the total amount of energy that is consumed by network nodes to perform the whole data aggregation procedure. It focuses on measuring consumed energy for establishing the MA migration infras-

Figure 3-8: The number of isolated nodes.



(a) small area $(200\times200)m^2$

(b) medium area $(400\times400)m^2$



(c) large area $(800\times800)m^2$

Key points of the figures:

1. The number of isolated node is – increased – when area size increases.
2. Isolated node count is reduced when node count increases.

tructure, routing the MAs and network deployment and maintenance [Wu et al., 2004]. Energy is a critical issue in WSN as sensor nodes are usually powered by a couple of AA batteries. Changing and/or re-charging the nodes batteries is not easy as the network may be deployed in hostile and inaccessible environments. For this reason, the objective of WSN protocols is to reduce energy consumption as much as possible to maximise the network lifetime. Intelligent power consumption and minimising network overhead (computation and communication) are two key issues that need to be considered to achieve this objective.

2. **Total number of captured data samples (accuracy):** represents the number of data samples that are properly collected and reported to the sink. This parameter is rooted in the routing algorithm's ability to find data regions and establish reliable links to forward MAs for data aggregation and return the aggregated result to the sink. For example, a protocol is more effective if a greater number of data samples are captured, aggregated and delivered to the sink. This parameter has a high impact on the robustness of data aggregation. This means that maximising the number of data samples in the data aggregation procedure may result in giving a more precise summary value of data samples to the data consumer. Hence, the consumer would be able to make better decisions on the collected data. The total number of captured data samples is called the accuracy of data aggregation routing in this thesis [Boulis et al., 2003].

3. **Average data collection end-to-end delay:** this parameter represents the average End-To-End delay (ETE) of MAs during the data aggregation procedure. It is measured as average time since the MAs start to collect data until they return to the sink and deliver the results [Chen et al., 2009b], [Chen et al., 2007]. ETE depends on communication and computation delays at the sensor nodes such as medium access, message reception and transmission, itinerary planning and aggregation delays. This parameter influences data accuracy and freshness [Solis and Obraczka, 2004]. This means that data samples may be expired or lose their meaning when they are delivered to the sink late. Hence, minimising the delay maximises accuracy and freshness of collected data samples.

4. **Total MA hop count:** this parameter is collected in order to measure the routing protocol's ability to establish shortest (minimum hop count) paths for MA to migrate [Chen et al., 2009a], [Wu et al., 2004]. It is measured as the total number of hops that are visited by the MAs during data aggregation procedure. This parameter depends upon the intelligence of the routing protocol in forwarding MAs through optimal (shortest) paths and avoiding random walks and/or blind mi-

grations. It influences the network energy consumption and end-to-end delay. In other words, the network resource consumption and end-to-end delay is reduced if the MAs are transmitted through minimum hop count routes. On the other hand, establishing random links for MA to migrate increases route hop count that results in increasing communication cost and delay. Owing to this, efficient routing protocols aim to reduce the hop count (by avoiding random and/or blind MA migrations) to reduce data collection delay and save network resources.

5. **Total transmitted traffic:** this represents the amount of transmitted (sent and received) network traffic during the data aggregation routing procedure [Basurra, 2012]. The network traffic includes: transmitted control packets and MA. The control packets are transmitted by network nodes to deploy the network and/or discover, establish and maintain the routes. Routing control packets include: Hello, route request/reply, route errors and maintenance, routing updates and acknowledgements. Network resource consumption is increased if transmitted information is increased. Moreover, data collection delay is increased as increasing the number of control packets results in increasing network traffic. In fact, increasing control packets results in increasing buffering, wireless channel access and transmission delays. Hence, reducing the transmitted network traffic results in a reduction of energy consumption and delay.

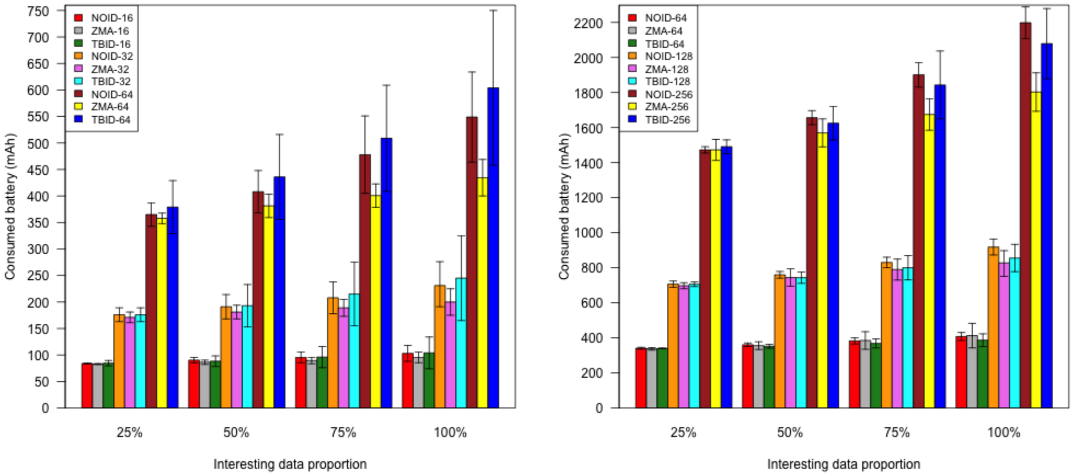
3.4 Results

This section evaluates the performance of ZMA, NOID [Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010] according to the chosen routing performance parameters that are described above. Each parameter is measured for the proposed (ZMA) and the benchmark (NOID, TBID) protocols in 36 different scenarios according to three different areas, three different node counts, and four data densities (proportions of interesting data sample) in the network.

3.4.1 Total Energy Consumption

Introduction: Power consumption is a vital parameter for WSN protocol design as the sensor nodes are highly energy restricted. It is essential to save energy consumption in routing protocol as much as possible to maximise the network lifetime. This section evaluates the energy efficiency of ZMA as compared to two benchmark protocols namely NOID and TBID. A protocol has a better performance if it reduces the energy consumption.

Figure 3-9: Energy consumption of MA routing protocols.

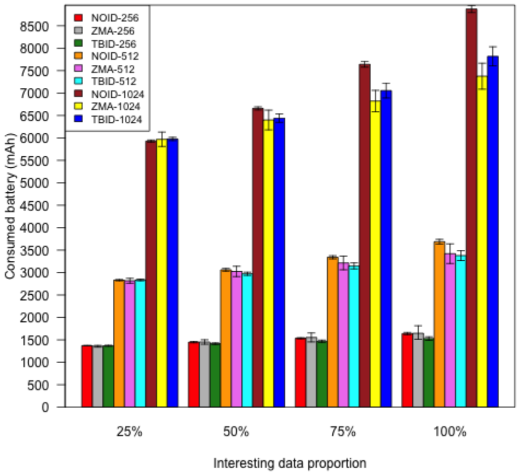


(a) small area (200×200)m^2

(b) medium area (400×400)m^2

Key points of the figures:

1. ZMA is more energy-efficient than the benchmark protocols.
2. Increasing the area size reduces energy-efficiency of ZMA.
3. Increasing node count and/or data density enhances energy-efficiency of ZMA.
4. TBID outperforms NOID when data density and area is increased.



(c) large area (800×800)m^2

Evaluation: From figure 3-9, it is observed that ZMA reduces the energy consumption as compared to NOID and TBID when network node count increases. In addition, ZMA outperforms the benchmark protocols in terms of energy consumption when data density increases, especially in dense networks. ZMA does better than NOID and TBID for two reasons:

1. **Limiting communications:** ZMA limits the routing communications into the data regions. This means that the nodes communicate to each other if they belong to the same zone and have data samples which match the MA requirements. Otherwise, the nodes leave the communications and go to sleep to save energy. On the other hand, the nodes broadcast messages to discover/establish MA routes in NOID. Hence, NOID suffers from overhearing as the network is flat and any node residing in the radio range of a sender would receive the routing messages during network deployment and/or MA itinerary planning.
2. **Intelligent MA migration:** in ZMA the MAs are routed to the nodes that have greatest connectivity to the interesting source nodes. ZMA forms the MA routes in a data centric manner (DC) in which each link is allocated by a weight value according to the DC connectivity degree and distance to the centre of event regions. Using the weight value, the MAs are guided via shortest paths to the nodes which are connected to a greater number of interesting source nodes. In addition, ZMA localises the MA migrations into the data regions to avoid blind and/or random migrations. This means that MAs move at each data region only if an interesting data sample is waiting to be collected. This results in reduction of MA migration hop count and consequently energy consumption. On the other hand, the benchmark protocols do not use DC regions to visit the source nodes. They forward MAs through the links which are locally selected in a greedy manner. The MAs select the minimum cost/delay link amongst all available paths in the single-hop neighbourhood for each migration. It does not give the MAs a broad enough view to consider other available source nodes and data regions which reside beyond the single-hop neighbourhood. Hence, the MAs need to blindly migrate between different event regions to visit the discovered source nodes that reside on the migration routes. In consequence, the number of MA migrations is increased resulting in higher energy consumption.

According to figure 3-9, ZMA energy consumption increases when the area increases. In other words, ZMA is more energy efficient in comparison to the benchmark protocols if it is used in a small area such as 200×200 m^2 for data aggregation. This is because the number of zones and data regions is increased when the area increases. Data regions are formed in ZMA by the source nodes which are connected via singe-hop links. Hence, the number of data regions and consequently routing overhead (control packets and/or MAs) is increased when the area size and/or node count increases. In addition, increased energy consumption in a large area reflects a trade-off between the energy consumption and accuracy. Energy efficiency of ZMA reduces as compared to NOID and TBID when the area size is increased because of ZMA's capability to find and capture a greater number of desirable data samples in the network. This means that enhanced accuracy increases the energy consumption in ZMA when the area size increases. On the other hand, the inability of NOID and TBID to provide efficient MA itinerary plans to find and capture interesting data samples does not increase energy consumption in large and/or dense networks. Energy-Accuracy correlation will be discussed in chapter 5.

Comparing the benchmark protocols, it is observed that TBID outperforms NOID when the area increases. The MA infrastructure (tree) for data aggregation is the reason why the energy efficiency of TBID improves (as compared to NOID) in large networks. This means that the routing overhead is reduced in TBID due to limiting the routing communications (i.e control packets) in the nodes which reside on the MA migration infrastructure. That is, the nodes residing on the MA migration infrastructure use unicast or multicast instead of broadcast to transmit the control packets to route the MAs, resulting in reduction in message overhearing and network energy consumption. On the other hand, NOID needs to broadcast control packets for MA routing and/or migration as the network is flat and there is no infrastructure to route the MAs. In addition, blind MA migrations increase the energy consumption in NOID. The MAs need to return to visit missed or non-visited source nodes, as NOID gives the MAs a single-hop view of the network for each migration. This increases MA hop count and consequently energy consumption, especially when the network is large and dense or the number of event regions is high.

3.4.2 Total Number of Captured Data Samples (Accuracy)

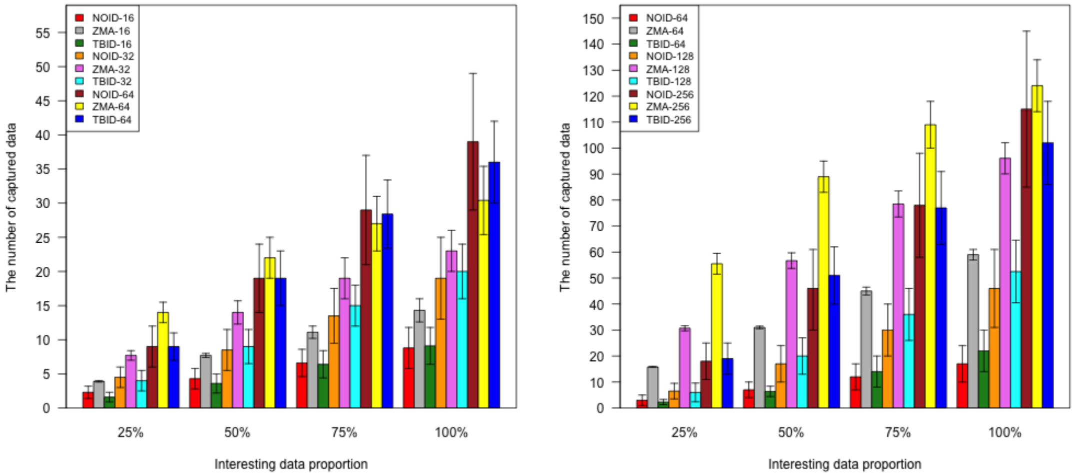
Introduction: We define accuracy as a measure of the number of interesting data samples that are collected from the nodes and reported to the sink by the MAs. Maximising the accuracy is the objective of data aggregation routing as it gives data consumer the ability to making better decisions based on the aggregated result. This section evaluates the accuracy of ZMA in comparison to TBID and NOID.

Evaluation: From figure 3-10, it is apparent that ZMA outperforms both the benchmark protocols in terms of accuracy. The MAs in ZMA have the ability to find source nodes and deliver captured data samples to the sink in either sparse or dense networks. According to the results, TBID and NOID are not efficient to find and capture desirable data samples when the network is sparse or data density (the proportion of desirable data) is low. In other words, the accuracy of the benchmark protocols is highly dependent on the node count and/or data density in the network. The reason is that the MAs are not informed by the intermediate nodes about the location of source nodes at which to gather data samples. In TBID and NOID, the event regions and/or source nodes can be discovered by the MAs if the intermediate nodes provide the required routing information for the MAs to move. The protocols are able to find the event regions in sparse networks as the interconnectivity between the intermediate/source nodes to provide the required routing information for the MAs is lower. On the other hand, increasing the interconnectivity between the network nodes sharply improves the accuracy of the benchmark protocols as the node count and proportion of desirable data increases.

The accuracy of ZMA is better than TBID and NOID as the area increases. This is for three reasons:

1. **Forming data regions:** ZMA has the ability to discover and form the event regions for the MAs to migrate. The MA migration areas are formed in a DC manner by the sensor nodes that have interesting data to report. It would result in interconnecting the source nodes for the MAs to move and visit. Hence, the MAs would be able to visit a number of source nodes which are connected through single or multi-hop DC links if one of them is visited. However, the benchmark protocols do not form event regions but move the MAs through chain (NOID) or tree (TBID) routes that are established according to a greedy algorithm which considers only the link cost. The MAs select the source node which has the minimum link cost to move without considering the data regions or existing connections between the source nodes. Thus, there is a probability of missing out source nodes which do not reside on minimum cost links. This could lead to reduced accuracy.
2. **Bottom-up MA migration:** ZMA utilises a bottom-up scheme for MA

Figure 3-10: Accuracy of MA routing protocols.

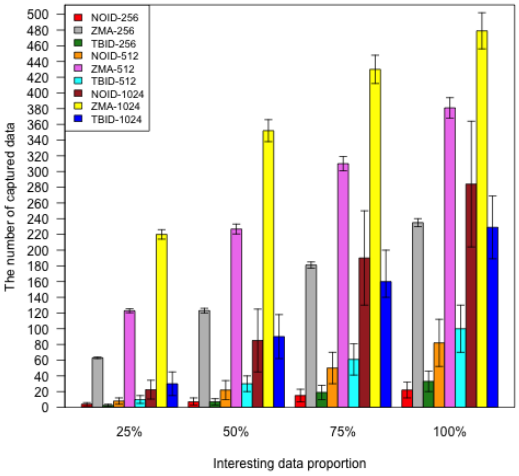


(a) small area $(200\times200)m^2$

(b) medium area $(400\times400)m^2$

Key points of the figures:

1. ZMA achieves a greater accuracy in comparison to the benchmark protocols.
2. Increasing the area size enhances data aggregation accuracy in ZMA.
3. ZMA under-performs the benchmark protocols if the network deployed is small and dense.
4. TBID is better than NOID if network is sparse, whereas NOID outperforms TBID when network is dense.



(c) large area $(800\times800)m^2$

migration. This means that the MA migrations are started from the centre of event regions (ZMACs) that are surrounded by the desirable source nodes. Each MA migration is an opportunity to capture one new data sample as it starts from a ZMAC node that is close to the centre of an event region and has short links to desirable source nodes. On the other hand, the benchmark protocols start MA migration from the single-hop neighbours of the sink (top-bottom scheme) that are aware of the event regions. Hence, the MAs miss out a set of source nodes that are not able (or fail) to join the routing infrastructure or inform the MA dispatchers (single-hop neighbours of sink).

3. **Maintaining the list of non-visited source nodes:** ZMA records the address of visited and non-visited nodes to avoid looping and to visit missed source nodes. Using the list, the MAs may return to MP nodes which have links to the non-visited source nodes if any source node is missed to visit. This results in increased accuracy, especially when the network is dense.

NOID outperforms TBID when the data density is high, whereas TBID slightly has a better performance (in terms of accuracy) when the number of interesting source nodes is low. This stem from increasing the number of source nodes which fail to join the tree infrastructure in TBID due to routing and/or tree establishment message (control packets) conflicts when the network is dense. On the other hand, increasing the network interconnectivity (node density) in NOID would result in increasing the number of source nodes that are able to provide the required routing information for the MAs which reactively move to collect the data samples. This results in increasing the accuracy in NOID as compared to TBID when the network deployed is dense. Conversely, TBID outperforms NOID when the network is sparse. This is because TBID collects the data samples through a tree infrastructure which is formed by desirable source nodes. The tree infrastructure increases the interconnections between the source nodes when they are sparsely scattered in the network. Hence, the number of visited source nodes and accuracy is increased when the network is sparse.

According to figure 3-10a, it is observed that the accuracy of TBID and NOID is better than ZMA in a dense network with a high number of source nodes in a small area. In this case, a smaller number of event regions is formed in ZMA, resulting in a small number of MAs for data collection. On the other hand, a greater number of MAs is generated in NOID and TBID because of an increased number of source nodes which are to communicate directly with the sink or its single-hop neighbours in a small area. Figure A-1 (in Appendix A) shows the number of

MAs generated by the protocols respectively in small, medium and large networks. Increasing the number of MAs and/or the node count (which leads to increased interconnectivity between the source nodes) increases the probability of finding and capturing desirable data and consequently improves the accuracy in TBID and NOID over ZMA in a dense network in a small area.

3.4.3 Average Data Collection End-to-End Delay

Introduction: Average end-to-end delay is the average delay from when the MAs begin data aggregation until they return to the sink. Reducing ETE is an objective of data aggregation as it can enhance data freshness. Collecting data samples and reporting the results to the sink as quick as possible enables the system to provide fresh data for analysis.

The average ETE of MA data aggregation routing is influenced most by two parameters as below:

1. **MA itinerary planning delay:** this delay is caused by the nodes to discover and establish the MA migration routes. It depends upon the proactivity/reactivity of the routing algorithm to plan the MA itineraries using the routing information, such as distance to the next hop or link cost.
2. **The communication delay:** the wireless communication delays depend on the distance between the sender and receiver, the number of intermediate hops, network traffic, message size and node characteristics (i.e operating system and/or queue size/delay) [Ardakani et al., 2014]. Increasing the path hop count has the potential to increase end-to-end delay as a greater number of intermediate nodes would participate in routing to send/receive control packets/MAs. Hence, ETE is reduced when the MAs migrate through short routes with minimum hop count paths.

Evaluation: Figure 3-11 shows that ZMA reduces the average ETE as compared to NOID and TBID, especially when the node count increases. There are three reasons for the reduction of end-to-end delay in ZMA:

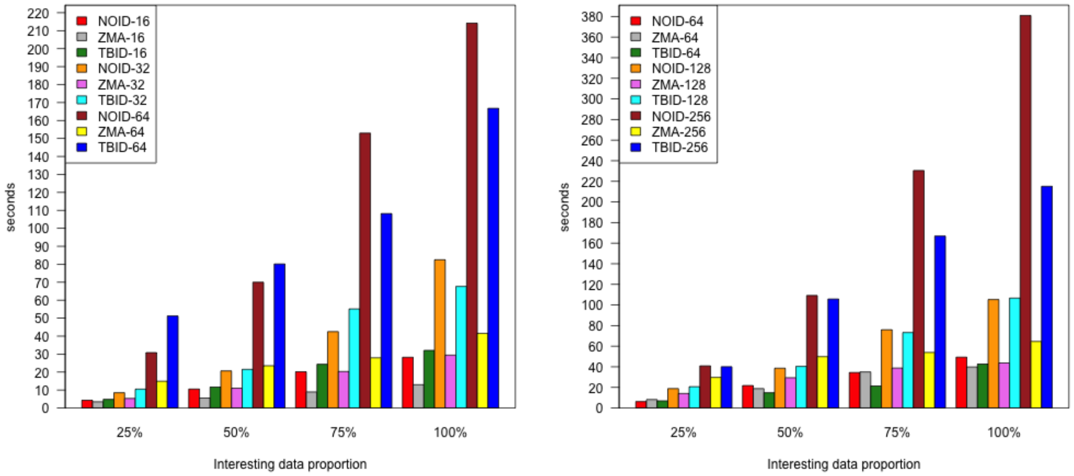
1. **Avoiding blind and/or random MA migrations:** ZMA move the MAs via the paths that are established according to a weighting function which considers the nodes DC connectivity degree and distance to the event sources. The paths locally interconnect the source nodes and form data regions to move MAs if any desirable source node is available to visit. Owing to this, the MAs avoid unnecessary, blind and/or random migrations to find

the location of event regions. This results in decreased path hop count and ETE. On the other hand, path hop count is increased in NOID and TBID as the MAs blindly move from the sink (single-hop neighbours) until the event regions are found heuristically. The MAs do not have a broad view about the event region locations and/or DC links between the source node as they use local routing information (i.e single-hop link cost) to move. Their migration hop count is increased because they find the source nodes heuristically (NOID) or blindly move through the routing infrastructure to capture data samples (TBID), especially when the network is dense.

2. **Increasing parallelism degree:** ZMA reduces ETE compared to TBID due to a higher degree of data aggregation parallelism from using a greater number of MAs at data regions (figure A-1 in Appendix A). Each data region (centred at a ZMAC) has a MA which is in process of collecting and aggregating intra-region data samples. The MAs move in parallel via the intra-region paths which are already computed proactively using the weighting function. This reduces ETE in ZMA.
3. **Hybrid routing:** ZMA utilises a hybrid routing scheme in which MAs move via proactively created links at each data region and then reactively establish routes to the sink. The objective is to reduce MA itinerary planning delay caused by reactive routing at data regions. Reactive routing increases ETE as the MAs need to collect the required routing information at each node to establish the migration paths. According to figure 3-11, ETE is significantly increased in NOID (reactive) especially when the node count and area increases. Increasing the number of multi-hop paths in large networks increases ETE in reactive routing as a greater number of nodes would participate in collecting/reporting routing information for the MAs. For this reason, ZMA moves a number of MAs, which move in parallel via proactive links to capture/aggregate data samples at data regions. This results in reduced itinerary planning delay and consequently decreased ETE.

ZMA has increased average ETE when the network is large and sparse. This is because of the MA itinerary planning delay to find the desirable source nodes in sparse networks. This means that ETE is increased in ZMA as the MAs have the ability to collect and aggregate a greater number of data samples when the network is sparse (see figure 3-10). On the other hand, the benchmark protocols such as TBID are able to find/visit a fewer number of source nodes when the nodes have a low degree of connectivity. This results in reductions to ETE as

Figure 3-11: End-to-end delay of MA routing protocols.

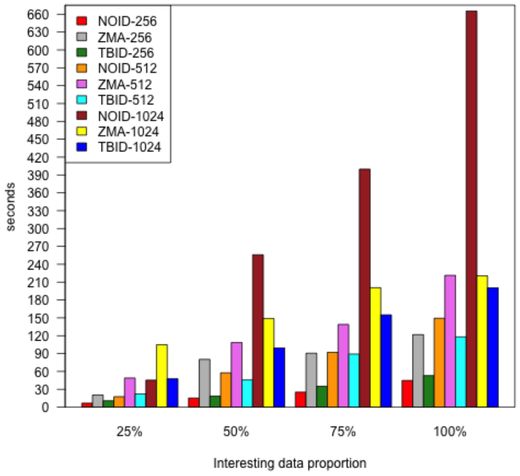


(a) small area (200×200)m^2

(b) medium area (400×400)m^2

Key points of the figures:

1. ZMA outperforms the benchmark protocols when node count increases.
2. ETE is increased in ZMA when the network is large and sparse.
3. TBID is better than NOID if data density is high, whereas NOID outperforms TBID when data density is low.



(c) large area (800×800)m^2

the data aggregation procedure is finished sooner and the MAs return to the sink sooner.

As figure 3-11 shows, average ETE is increased in NOID (as compared to TBID) when the data density is high, whereas it is reduced when the number of source nodes is low. Routing scheme (reactive or proactive) and the ability to discover source nodes in a sparse network are two reasons as explained below:

1. **Routing scheme:** TBID utilises proactive routing to move the MAs for data aggregation over the tree infrastructure. As the itinerary planning delay is lower in proactive routing than reactive, ETE is decreased in TBID (especially if the network is dense and large) compared to NOID. In addition, the impact of routing scheme (proactive vs. reactive) on ETE is observed in TBID as compared to ZMA. According to figure 3-11c, TBID reduces end-to-end delay as compared to ZMA when the area increases. This is because of establishing paths reactively in ZMA to forward/return the MAs to/from the data regions to the sink. When area increases, the MAs need to establish multi-hop paths to the sink via a higher number of intermediate nodes (TS). Hence, the MAs reactively collect the routing information (i.e available links to TS nodes at the next inner zone) over longer (in terms of hop count) paths when they move back to the sink. Hence, ETE is increased.
2. **Source node discovery in sparse network:** TBID is able to find a greater number of source nodes (as compared to NOID) when the network is sparse and the number of source nodes is low (fig. 3-10). This results in increasing the end-to-end delay (ETE) in TBID for collection and delivery of data samples.

3.4.4 Total MA Hop Count

Introduction: MA hop count is measured by the total number of nodes that are visited by the MAs during the data aggregation routing procedure. The objective is to minimise the MA hop count as it affects ETE and energy consumption. In other words, the MA hop count should be minimised to save energy and improve data freshness. Non-optimal itinerary planning, blind migration and random walks are the key issues that can affect MA hop count.

Evaluation: According to figure 3-12, ZMA reduces the MA hop count as compared to the benchmark protocols when the node count increases. This is because of the ability of ZMA to avoid blind/random migrations and establish short paths to forward the MAs. ZMA forms data regions to localise MA migrations and minimise MA path hop count at each zone. The MA migration paths are established in a DC manner according to the weight values which are computed based on the distance to the centre of data regions. Hence, the MAs are moved if the next node is not visited and knows a data region or a set of non-visited source nodes. On the other hand, blind MA migration is the reason for increased MA hop count in NOID. It selects the next hop node (from a single-hop domain) based on the cost of link and without considering the location of event regions. Hence, the MA hop count increases as the MAs need to frequently move between variant parts of network to visit the source nodes.

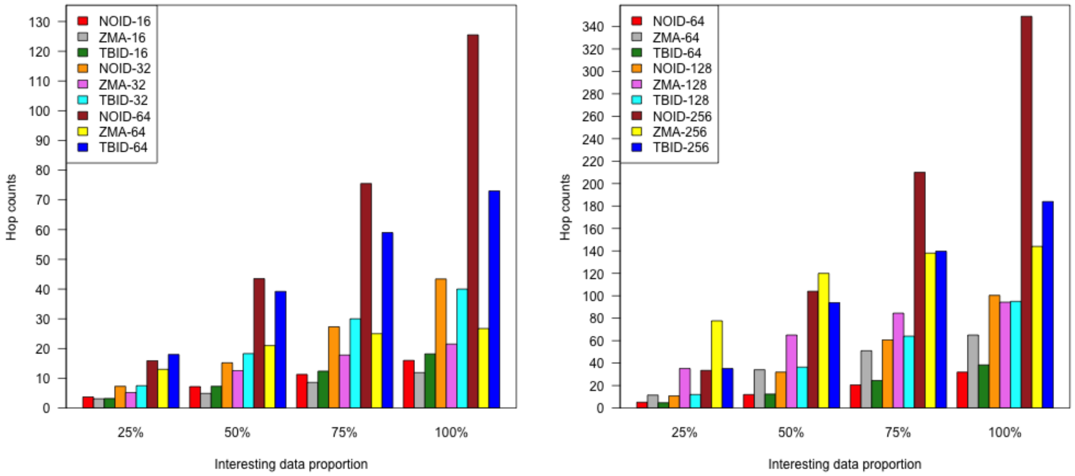
According to the results, the MA hop count in ZMA is increased in comparison to the benchmark protocols, when the network is sparse. This is because ZMA can find and capture a greater number of source nodes compared to the benchmark protocols. According to figure 3-10, the MAs need to move through longer routes to capture a greater number of discovered data samples in sparse networks. However, the MAs in NOID or TBID would return to the sink via paths with fewer hops as they are not able to efficiently find the data regions and capture the desirable data samples.

To compare MA hop count in benchmark protocols, it is observed that MA hop count in NOID is greater than TBID. This is because MA migration in TBID is limited to the nodes which reside on the tree infrastructure. Hence, blind migration is avoided and the MAs move if the destination nodes reside on the tree. On the other hand, the MAs consider a singe-hop area of the network in NOID to find the source nodes. This results in increased blind migration and consequently increased MA hop count.

3.4.5 Total Transmitted Traffic

Introduction: Total transmitted traffic is measured as the total amount of data transmitted (sent/received) by the network nodes. The network transmitted traffic comprises the MAs and control packets which are transmitted during data aggregation routing. The MAs are used to transmit data samples throughout the network, whereas the control packets are transmitted to establish the routing infrastructure, deploy the network, route MAs, update network topology and/or access wireless channels. Total transmitted traffic has a high impact on network energy consumption and ETE. This means that minimising the transmitted traffic should result in decreased network congestion, saving network energy resources

Figure 3-12: Hop counts of MA routing protocols.

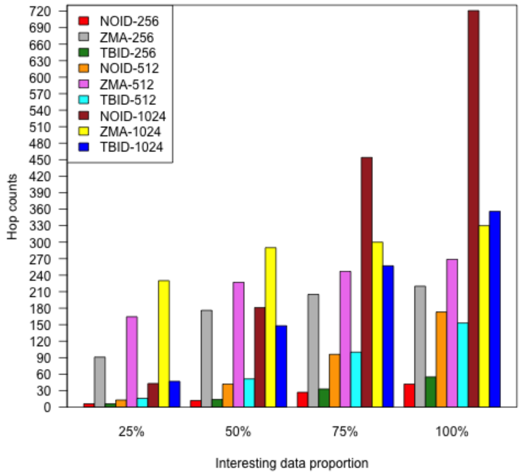


(a) small area $(200\times200)m^2$

(b) medium area $(400\times400)m^2$

Key points of the figures:

1. Hop count is reduced in ZMA if the network is dense.
2. The benchmark protocols move MAs less than ZMA when the network is sparse.
3. The MAs move more in NOID than ZMA and TBID.



(c) large area $(800\times800)m^2$

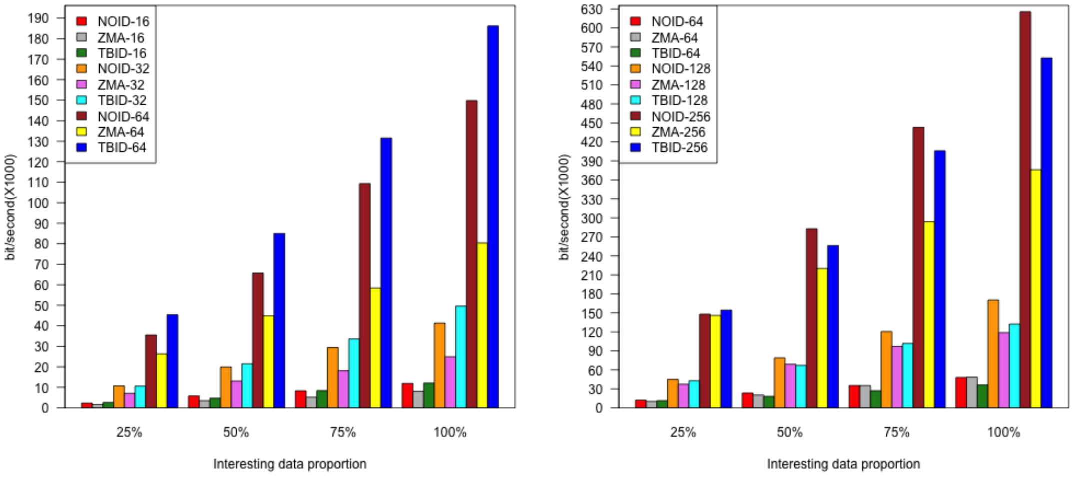
and reducing ETE. It is explained as below:

1. Energy consumption: transmitting network packets (i.e control messages) increases energy consumption at both sender and receiver sides of communication. For example, energy consumption is increased when a routing protocol transmits a great number of control packets to discover/establish routes and forward data packets/MAs.
2. Network congestion: network congestion can be increased if transmitted network traffic increases, as a greater number of nodes need to access the wireless channel to send/receive the messages. Network congestion reduces the network throughput due to increasing the message failure rate.
3. Communication delays: increasing network traffic results in increasing communication delays. Sensor nodes need to queue the messages until the wireless channel become available (idle listening) or re-transmit the messages in the case of congestion. This increases end-to-end delay of data collection especially when network is dense.

Evaluation: From figure 3-13, it is apparent that ZMA has less total transmitted traffic as compared to TBID to NOID especially when the node count increases. The reason is that ZMA localises the network transmissions into the network zone and/or data regions. For example, control packets are transmitted (in multicast) between the nodes which reside in the same zone instead of any node which resides in the radio range of the sender node (broadcast). Hence, ZMA reduces the overhearing of the network transmissions by limiting the control communication to the zones. However, the total transmitted traffic is slightly decreased in TBID (as compared to ZMA) when a sparse network is deployed in a large area. For example, according to figure 3-13c, total transmitted traffic of TBID is less than ZMA when a network with 256 nodes is deployed in a 800×800 m^2 area. TBID transmissions are limited to the nodes which reside on the tree infrastructure. The number of nodes which join the tree and participate in transmitting the network traffic reduces when the network is sparse deployed. This results in a reduction in the number of transmitted control packets. In addition, NOID (and TBID) capture and transmit a fewer number of data samples (according to figure 3-10c) when the network is sparse because it is not efficient (due to lack of required routing information at the nodes for the MAs to move) to find the RS source regions. Hence, the MAs transmit less data through the network. This results in lower transmitted traffic in NOID (and TBID) as compared to ZMA.

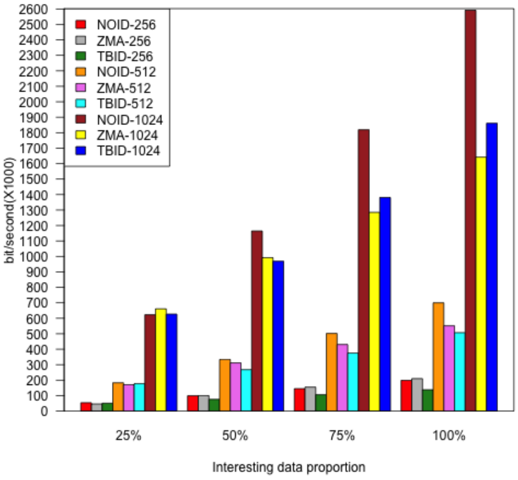
According to figure 3-13, the total transmitted traffic is increased in NOID as

Figure 3-13: Transmitted network traffic of MA routing protocols.



(a) small area (200×200)m^2

(b) medium area (400×400)m^2



(c) large area (800×800)m^2

Key points of the figures:

1. ZMA has less network traffic when the network is dense.
2. Network traffic in ZMA is slightly increased (compared to TBID) when the network is sparse and large.
3. Network traffic in NOID is increased when area increases.

compared to TBID when the area increases. In large networks, a greater number of source nodes must communicate through multi-hop links. It results in increased transmitted traffic (caused by overhearing), as the number of nodes which participate in routing increases. On the other hand, the network communications would be limited to the nodes which reside on the tree infrastructure in TBID. In consequence, total transmitted traffic is reduced because the number of node which participate in routing are limited to the desirable source nodes that join the tree infrastructure and not all network nodes.

3.5 Discussion and Conclusion

ZMA performs well compared to NOID and TBID in terms of energy, accuracy and delay especially as the area and the node count increases. It forwards the MAs to find, capture and aggregate desirable data samples from the source nodes which are scattered in ER (100% detection) or RS (random detection) model. The performance of ZMA does not depend on the data heterogeneity in the network. This means that ZMA is able to find, collect and report the desirable data samples from the sensor nodes which are able to measure variant data types. Forming data regions and communication zones, utilising bottom-up MA migration approach, hybrid routing and establishing weighted DC links to forward MAs are the key techniques which are used in ZMA to enhance the data aggregation routing performance.

Energy, accuracy and delay are the data aggregation routing metrics that have been identified and measured. In this research, the objective is to maximise data aggregation accuracy while minimising the network energy and end-to-end delay. Energy, accuracy and delay is influenced depending on the single/multi-hop communication, proactive/reactive routing, node density and data availability. These are summarised as follows:

1. Energy consumption: energy saving is a vital requirement for WSN as it is determines network lifetime. Indeed, reducing energy consumption enhance network availability to sense and report over longer periods. Energy consumption is in a trade-off with accuracy, as collecting a greater number of data samples to increase the accuracy may result in increased energy consumption.

2. Accuracy: data aggregation accuracy should be maximised due to data consumer requirement. A WSN is established to measure and report the ambient data for the data consumer. Hence, increasing the number of data samples which participate in data aggregation procedure enhances the robustness of collected

data. Accuracy is in a trade-off with data aggregation delay as collecting a greater number of data samples takes longer time to provide the aggregated result at the sink.

3. End-to-end delay: this is a QoS metric that is required to be minimised in data aggregation routing protocols. Indeed, reducing the delay has the potential to increase data freshness and better support real-time applications. It is in a trade-off with energy consumption. Although delay is reduced over the minimum hop count paths, energy consumption is increased depending on the euclidean distance between the nodes.

The (anti-)correlations between energy consumption, accuracy and delay form a triangle in which each edge focuses on a potential correlation between the two metrics. First, energy-delay focuses on minimising energy consumption while reducing delay. Second, energy-accuracy considers maximising the data aggregation accuracy if consumed energy is minimised. Third, delay-accuracy studies the correlations between delay and accuracy to minimise the end-to-end delay while maximising the data aggregation accuracy. Chapter 5 discusses these correlations in detail.

From the results, it is apparent that ZMA outperforms TBID and NOID in terms of energy consumption. Forming data regions and communication zones are the techniques which reduce the energy consumption in ZMA. They are explained as below:

1. **Data regions:** data regions are formed to reduce the MA migrations by connecting the desirable source nodes in the network. This technique allows the MAs to avoid random/blind migration and limit the MA search domain to find the data samples. Hence, the MAs are able to capture and aggregate desirable data samples by minimising the migration path hop count. On the other hand, NOID does not consider data regions, but selects the minimum cost link to the single-hop neighbour which has interesting data or a link to a desirable source node. Figure 3-12 shows that NOID hop count is sharply increased when the node count and/or area increases, resulting in increased energy consumption.

2. **Communication zones:** they limit the network communications to the nodes which reside in the same zone and use control packets to establish data regions or DC links. On the other hand, overhearing rises if the communications are not limited to particular regions and any node in the sender radio range stays available to receive the forwarding packets. Overhearing and increased transmitted control packets are the reasons of increased energy consumption in the benchmark protocols according to Figure A-2 in Appendix A. However, increased energy consumption in ZMA in comparison to NOID and TBID when the network is sparse

is because of the inability of TBID and NOID to capture data samples when a low proportion of desirable source nodes in a sparse network. In other words, ZMA's higher energy consumption is due to the accuracy-energy correlation which is explained in Chapter 5.

ZMA sharply increases the data aggregation accuracy when the area and/or node count rises. Forming data regions and bottom-up MA migration techniques result in increasing the accuracy of ZMA, as follows:

1. **Forming data regions:** This aspect increases the chance of capturing data as finding one source node from a data region may result in visiting a set of source nodes which are connected via DC links and have interesting data to report. DC links forward the MAs to the next source nodes according to a sequence (the weight value) which is calculated according to the connectivity degree of each host source node. This means that a MA selects a node to move to next if it knows a greater number (DC connectivity degree) of non-visited desirable source nodes in comparison to all candidate nodes. On the other hand, NOID under-performs ZMA in terms of accuracy as it does not establish the DC links to connect the desirable source nodes. NOID considers the link cost of source nodes which are connected in single-hop to forward the MAs without considering DC connectivity degrees of the nodes. This results in random/blind migration and reduced data aggregation accuracy.

2. **Bottom-up MA migration:** This technique enhances data aggregation accuracy in ZMA by forwarding the MAs from the centre of data regions to the sink for data aggregation. Each MA starts its journey from ZMAC (centre of data regions) to capture the data samples from the respective data region and then reactively establish a path to the sink to deliver the results. This increases data aggregation accuracy as a MA journey starts from a node which is connected to a set of desirable source nodes. Hence, each MA migration is a chance to capture a data sample. On the other hand, TBID and NOID utilise top-down MA migrations which forward the MAs from the single-hop neighbours of the sink to the source nodes in the event regions. The lack of DC connections between randomly distributed source nodes reduces the data aggregation accuracy when top-down approach is utilised. This means that the MAs miss a set of source nodes to visit if they are not connected through either single or multi-hop links to the single-hop neighbours of the sink or routing infrastructure (tree or chain). Owing to this, data aggregation is significantly reduced especially when interesting source nodes are sparsely distributed in a large area (according to figure 3-10c). However, the

accuracy of NOID and TBID improves if the network is dense and small. In this case, top-down MAs are able to find the source nodes and capture data samples as a greater number of source nodes are able to communicate directly to the sink or its single-hop neighbours. According to figure 3-10a, the MAs are able to capture a greater number of data samples from the source nodes which are reside on connected migration paths such as chain (NOID) or tree (TBID) as compared to ZMA.

ZMA reduces the data aggregation delay as compared to NOID and TBID. This is because it decreases the MA hop count and utilises parallel proactive data aggregation. These are explained as follows:

1. **Decreasing hop count:** ZMA reduces blind/random migration and consequently end-to-end delay as the MAs migrate through DC links which are established between desirable source nodes. This results in avoiding random/blind MA migration and consequently reduces the MA hop count (according to figure3-12), especially when the network is dense and large. Decreasing MA hop count reduces routing communication delays and consequently decreases data aggregation ETE.
2. **Parallel proactive routing in data regions:** ZMA moves the MAs in parallel via proactive intra-region links to capture/aggregate data samples at data regions. This results in increased data aggregation parallelism and reduced itinerary planning delays compared to reactive routing. On the other hand, reactive routing (NOID) sharply increases the delay as the MAs need to wait at each node until the required routing information to compute the migration path is collected. This increases itinerary planning delay and ETE especially when the node count increases.

It is concluded that generally ZMA has a satisfactory performance in large and small network with a range of node counts and data densities. ZMA is intelligent enough to find the source nodes which are scattered in the sensing area either in RS(random detection) or ER(100% detection) model. It has the ability to collect and aggregate data samples over a heterogeneous network with variant data type sensor nodes due to utilising bottom-up MA migration scheme, and establishing data regions and weighted DC links. ZMA has a better performance in terms of energy consumption, end-to-end delay and accuracy as compared to the benchmark routing protocols, NOID and TBID, especially when the network is large and dense.

3.6 Summary

In this chapter, a mobile agent data aggregation routing protocol (ZMA) has been introduced. The objective of proposing ZMA is to maximise energy efficiency and data aggregation accuracy, and minimise the end-to-end delay. The simulation results show that the ZMA has a satisfactory performance, and that it satisfies its objectives as demonstrated by the comparison of the results for NOID [Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010]. ZMA aims to collect and aggregate data samples regardless of the event source distribution model (RS or ER). It has the potential to work over sensor nodes which are equipped with multiple sensing modules (i.e TelosB node [Williams, 2014]) to measure several kinds of event. ZMA reduces energy consumption as it avoids blind/random MA migrations to discover, collect and aggregate data samples over randomly established networks. In addition, forming data regions to reduce MA migration hop count and establishing communication zones to limit overhearing result in energy saving in ZMA. The end-to-end delay of data aggregation is lower in ZMA by reducing MA hop count, avoiding blind/random migration, parallel data aggregation and utilising a hybrid routing scheme. In addition, ZMA increases accuracy by forwarding MAs from the centre of data regions in a bottom-up migration scheme to collect and aggregate data samples.

Chapter 4

Client/Server Data Aggregation Routing

This chapter focuses on data aggregation in WSN according to the client/server model of routing. Client/server routing forwards data packets from the data regions to the sink through single or multi-hop paths which are formed over a flat or hierarchical infrastructure. Depending on the routing infrastructure, the intermediate nodes may perform in-network data aggregation to collect and combine the data packets. A client/server routing protocol for data aggregation is proposed in this chapter. It partitions the network into a set of data centric (DC) clusters to collect the data samples from the event regions. The collected data samples are aggregated at the Cluster Heads (CHs) and then the results are forwarded to the sink via a spanning tree.

The chapter is organised in seven sections. Section 4.1 introduces client/server data aggregation routing and then highlights the research points which need to be considered in this field of computer science. Section 4.2 provides a brief literature review for WSN clustering. It describes the Hamming distance technique as a potential approach to form data centric clusters in WSN. Section 4.3 introduces a hybrid client/server data aggregation routing protocol namely Cluster-Based Aggregation (CBA). This section focuses on the key techniques which are used in CBA to resolve the existing drawbacks and enhance the performance of client/server routing. Section 4.4 describes the experiment plan that is designed to test and evaluate the performance of CBA. Varying node count, area size and data density (the proportion of interesting data samples) are the factors that form the focus of the experiment scenarios. Section4.5 presents the result of the experiments according to the five metrics (total consumed energy, total number of delivered data samples, average end-to-end delay, average hop count and total transmitted traffic) normally used in the literature to evaluate data aggregation routing performance. It explains the effect of each parameter in evaluating the performance of CBA in comparison with two respected client/server routing protocols namely LEACH [Heinzelman et al., 2000] and Directed diffusion (DDiFF) [Intanagonwiwat et al., 2000]. Section 4.6 discusses and concludes the key points of the results to highlight the advantages and disadvantages of the proposed protocol. Finally, a summary of this chapter is provided in section 4.7 as a quick overview.

4.1 Client/Server Data Aggregation

There are five key issues that need to be considered by researchers and engineers in WSN client/server data aggregation routing:

1. **Energy consumption:** power resources need to be used efficiently in WSNs as they are highly constrained. Forwarding data packets over long paths, overhearing and message conflicts/collisions are the behaviours that increase energy consumption in flat WSNs. On the other hand, the cost of establishing and maintaining an hierarchical infrastructure must be minimised in hierarchical WSNs to save energy.

2. **Network congestion:** simultaneous access to the limited wireless channels increases network congestion and consequently enhance the probability of message failures in WSNs. It can increase network resource consumption as the source nodes need to re-transmit failed data packets. Network congestion is decreased in hierarchical networks, as compared to flat, due to reducing the number of nodes which need simultaneously to access the wireless channels. Hierarchical WSNs partitions the network into a set of groups in which a few number of nodes (group leaders/representatives) are in charge of managing the group communications. However, network congestion could be problematic in hierarchical WSNs as the number of group and/or leaders increases.

3. **Overhearing:** receiving network packets which do not belong to the receiver nodes increases network resource consumption in WSN. Hierarchical infrastructure has the potential to reduce overhearing (compared to flat networks) as the communications can be locally limited into the grouped nodes. Depending on the size of groups, however, overhearing is increased if the groups formed are large and/or dense.

4. **Delay:** end-to-end delay (ETE) should be minimised in data collection as it is key to data freshness. ETE would depend on network traffic and path length (hop count) from the source regions to the sink.

5. **Data collection/aggregation from ER(100% detection) or RS(random detection) event sources:** RS data collection increases network congestion, delay and resource consumption especially in a flat network, as each source node need to individually establish a path to forward data to the sink. It can be resolved in hierarchical networks by grouping the source nodes in which the group representatives forward the aggregated data of grouped nodes to the sink. It results in reduction of routing overhead, network traffic and resource consumption. However, the group leaders miss collecting data samples from source nodes which are not covered/joined by/to the hierarchical infrastructure. For this reason, the hierarchical infrastructure needs to minimise the establishment/maintenance cost and maximise coverage of event regions either in RS or ER.

In this chapter, a hybrid client/server data aggregation routing protocol namely Cluster-Based Aggregation (CBA) is proposed, described and evaluated. It forms a cluster-based infrastructure in a data centric fashion to collect, aggregate and report data samples. CBA dynamically partitions the network into a set of clusters based on the measured data using a lightweight clustering approach which is called the Hamming distance. The cluster-heads then form a Minimum Spanning Tree (MST) as the network backbone to forward aggregated results to the sink. A parallel collision-guided technique is used to minimise the establishment cost of the tree infrastructure. The performance of CBA is tested and compared to two conventional works: LEACH [Heinzelman et al., 2000] and Directed diffusion (DDiFF) [Intanagonwiwat et al., 2000]. The benchmark protocols are selected based on their similarities, maturities and/or popularities to investigate and highlight the advantages and disadvantages of both flat and hierarchical architectures in client/server data aggregation routing. CBA focuses on resolving the drawbacks of client/server data aggregation routing as follows:

1. Decreasing message overhearing and network congestion by limiting the data communications into data centric clusters.

2. Reducing end-to-end delay by forwarding data transmissions through minimum hop count links.

3. Reducing energy consumption by utilising lightweight techniques for establishing the routing infrastructure and forwarding data packets.

4. Supporting data collection in both models of event source distribution models (ER and RS) using data-centric routing techniques.

4.2 A Brief Review of Clustering

Clustering is a technique commonly used to establish hierarchical routing infrastructure in WSNs. Using clustering, the network is partitioned into a set of groups named clusters. The clusters can be formed based on nodes similarities in a set of distinctive features such as location, measured data and/or communication and computation abilities. The key benefit of utilising clustering technique in WSN is to reduce network congestion [Liu, 2012]. In a clustered network, data packet transmissions are forwarded from the source nodes to the cluster-head nodes, which are then responsible for managing the intra-cluster communications. Cluster-heads aggregate the collected data samples and forward the aggregated results to the sink via either single or multi-hop links. Owing to this, the number of nodes that simultaneously try to access the wireless channel is reduced. It would result in reducing the number of parallel transmissions and consequently the network congestion as compared to flat networks. Besides, reducing the number of simultaneous transmissions and network congestion increases data collection accuracy and scalability in clustered networks. This means that a greater number of data samples is received at cluster-heads for aggregation because the intra-cluster links are more reliable for the delivery of data packets in comparison to long routes which are established in flat network from the event regions to the sink.

To form the clusters, three attributes should be considered [Younis et al., 2006]: methodology, cluster properties and cluster-head selection. Clustering methodology is classified into three categories:

1. **Centralised:** the clustering information is collected at a single or multiple nodes (i.e sink) to optimally form the clusters. This scheme is expensive for WSNs as it needs frequent communications between the nodes to collect the required information and/or form/manage the clusters. Moreover, the performance of clustering would depend on the availability of the node(s) which is in charge of forming the clusters. The clustering procedure would fail if the node(s) fails or is not available. LEACH-C [Heinzelman et al., 2002] utilises a centralise scheme at the sink to form the cluster.

2. **Distributed:** allows for the local performance of clustering at each node without any centralised control. Although distributed clustering has the potential to resolve the drawbacks of centralised clustering, it forms overlapped or un-balances clusters. LEACH [Heinzelman et al., 2000] is an example of distributed clustering.

3. **Hybrid:** each cluster is formed in a centralised manner at a node (Cluster-Head), whereas the inter-cluster links are formed according to a decentralised

protocol. The contribution of hybrid clustering is to avoid forming overlapped and un-balanced clusters. TEEN [Manjeshwar and Agrawal, 2001] uses a hybrid approach for clustering.

There are four issues (clustering properties) that need to be considered in a clustering procedure [Abbasi and Younis, 2007]: cluster count, cluster size, inter-cluster and intra-cluster communications. The cluster count may be fixed or varied based on the node count and/or network application. It can be dynamically set by the network consumer in advance or during network deployment. For example, the cluster count is fixed in HEED [Younis and Fahmy, 2004], whereas it may be variable in LEACH [Heinzelman et al., 2000]. The cluster size can be either equal or unequal. The former establishes same size clusters (i.e CMLDA [Dasgupta et al., 2003]), whereas the latter forms clusters of variable size (i.e LEACH). Inter/intra cluster communications focus on the communication patterns used by the sensor nodes within or between the clusters. They can be formed either as single or multi hop, based on the network application, size and/or density. HEED [Younis and Fahmy, 2004] establishes the links either in single or multi-hop, whereas single-hop links are created in LEACH [Heinzelman et al., 2000].

Each cluster is managed by either one or multiple nodes, namely the Cluster Heads (CHs). They may play different roles as aggregator, relay and/or data collector to manage intra-cluster communications. Cluster-head selection is a challenging issue in clustering procedure as the efficiency and stability of a cluster depends on the CH properties. Selecting CH with low connectivity degree and/or residual energy leads to the formation of un-balanced, overlapped and/or unstable clusters. Hence, the nodes need to frequently or periodically re-cluster the network to adapt themselves with the application requirements or network characteristics. A CH is typically selected using one of three schemes [Jiang et al., 2009]:

1. **Deterministic:** the CHs are selected according to a pre-defined criterion such as location address in advance of network deployment.

2. **Adaptive:** CHs are dynamically selected based on a set of features such as residual energy and/or communication cost.

3. **Random:** the cluster-heads are randomly selected without considering any specific criteria.

Clustering has the potential to support contention-free MAC (Medium Access Control) protocols. Network MAC protocols are used to prevent interfering nodes from transmitting data packets at same time. Decreasing message collisions, network packet overhead, overhearing and idle listening are the key benefits of utilising MAC protocols in WSNs [Demirkol et al., 2006]. MAC protocols are classified into two categories: contention-based and contention-free. The interfering nodes compete to access the wireless channels in the former, but they are coordinated to access the channels without competition in the latter [Ye et al., 2002]. Contention-free MAC protocols allow the nodes to transmit data packets at a particular time or by a specific code without competition and/or additional overhead. Grouping the nodes into separate clusters allows the nodes to use contention-free MAC protocols such as Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) [Ali et al., 2006] efficiently. This means that clustered nodes are able to access the wireless channels at particular time slots (i.e TDMA) or by spreading communication code (i.e CDMA) without additional overhead and consuming network resources.

4.2.1 Hamming Distance

Hamming distance is a technique that is generally used to find the difference of binary values [Xiong et al., 2004]. This technique counts the number of flipped bits in fixed size binary data streams and returns the value of the difference as the distance. For example, the hamming distance of 101 and 000 is 2 due to the first and last bits which are different.

The Hamming distance can be used to cluster WSNs in a data-centric manner. The sensors nodes code the meaningful features of measured data such as data type, measuring time and/or location into binary vectors. Hence, the sensor nodes that have a lower Hamming distance are grouped into data-centric clusters in a distributed manner without requiring a centralised control [Arasu et al., 2006]. The benefits of utilising Hamming distance to form data-centric clusters in WSN are given below [Gasieniec et al., 2004], [Yao et al., 2007]:

1. Hamming distance has the potential to reduce computation overhead as it is not complex to perform. This technique focuses on encoding and decoding the selected features of measured data into the Hamming vectors.

2. Hamming distance assists in the formation of data-centric clusters. The clusters are formed by the nodes which have same or similar Hamming codes. It reduces network resource consumption for data aggregation routing as the communication links between the source nodes are formed according to the similarity of the measured data samples at each node. In other words, the nodes which have similar data types are interconnected via DC links. Network resource consumption is reduced when data packets are forwarded through data-centric instead of address-

centric links especially in data aggregation applications [Anastasi et al., 2009], [Krishnamachari et al., 2002].

3. This technique reduces communication overhead by reducing the size and/or number of messages which are transmitted during clustering or data collection procedures. Using Hamming technique, the sensor nodes transmit only a set of key features of original measured data (i.e data type) that are required to form the clusters. This clustering [Anastasi et al., 2009] reduces the overhead of routing by reducing the amount of transmitted traffic. Moreover, the number of transmissions is reduced as hamming distance technique has the potential to eliminate transmission of redundant data. As the Hamming distance technique allows the source nodes to find out the similarity of measured data samples, the source nodes would avoid transmitting same or similar data values which respectively have the same Hamming code or a close Hamming distance.

We use the ideas underlying Hamming distance in our new CBA algorithm given below.

4.3 The CBA protocol

This section proposes a cluster-based routing protocol (CBA) which supports data aggregation in a client/server model. CBA allocates a cost value to each node according to the distance and/or path hop count from the sink. Then, it partitions the network into a set of data-centric clusters using Hamming distance technique. The data packets are aggregated at the cluster-heads and hierarchically forwarded then through a spanning tree to the sink. The tree infrastructure is rooted at the sink and formed as a result of parallel route request collisions which are forwarded from the CHs to the sink.

4.3.1 Network Model

Similar to Chapter 3 (Section 3.2.1), the network model of CBA consists of three components: sink node(s), sensor nodes and event sources. The sink node(s) is/are the consumer access-point to monitor the network performance. Data samples need to be collected from the source nodes and delivered then to the sink for further processing. The sink(s) has/have no resource limitation in terms of power, storage, communication and computation ability.

The sensor nodes are static and can be homogenous or heterogeneous in terms of having variable levels/units of resources. They can be equipped with either a single or multi-sensor devices to sense a range of physical values such as light and/or temperature. The communication radio range of sensor nodes is R (constant for all the nodes). To manage message passing and wireless communications, it is assumed that the nodes are synchronised [Solis and Obraczka, 2004]. Each sensor node may play one or multiple roles according to its location in the network: Idle, Cluster Member (CM), Cluster Head (CH), Tree Member (TM) and Meet Point (MP). CMs are the source nodes which are clustered based on the type of measured data. CHs are the CMs that are selected at each cluster to manage inter/intra-cluster communications. TMs are the nodes residing on the tree infrastructure which is formed between the CHs and the sink. MPs are the TMs which are on joint routes from the CHs to the sink. TMs and MPs role and duty will be explained in section 4.3.4. Idle nodes do not perform any particular task in the network. They are usually isolated unless they are asked (by other nodes) to provide back-up services during network deployment, data collection and/or network healing.

The communication links in CBA are established in a data-centric manner. Data-centric routes are more desirable than address-centric ones especially for data collection applications in WSNs, as the measured data at the nodes is more important than node address (IDs) to establish a route [Krishnamachari et al., 2002]. The route formed from the sink to a source region is called a Forward Path (FP), while a route from a source node to the sink is called Backward Path (BP).

The event sources generate environmental data and can be either static or mobile. They may be scattered in the network in either Event-Radius (ER) or Random-Source (RS) models.

4.3.2 Cost Value Allocation

Each node in CBA is allocated by a cost value according to its distance (hop count) from the sink. The value is assigned to inform the nodes how far they reside from the sink. Using the cost value, each node would be able to guide network transmissions (i.e data or control packets) to either closer or farther regions from the sink. In fact, the nodes should avoid pure broadcasting, since that is usually utilised to forward network packets where a global addressing scheme and/or location information is not available. They utilise multicast or even unicast to forward network packets which need to be forwarded either to closer or farther areas from the sink. For example, the sink queriers are usually broadcast by each node throughout the network when the cost value technique is not used, whereas they are forwarded only to the nodes which reside in farther regions from the sink if the nodes are allocated by the cost values.

CBA utilises a similar approach to Energy Aware Routing [Shah and Rabaey, 2002] to allocate the cost values to the nodes. The procedure is started when the sink forwards the cost messages. The structure of cost messages is similar to $Hello_{v1}$ which is used by ZMA to create the zones in Chapter 3. As figure 4-1 shows, there are three fields in the header of message: sender Id, Hop Count(HC) and Total Cost (TC). HC shows the hop count and TC is the total distance value of the interconnected links on the forward path (FP). TC value is measured according to equation 3.4 using RSSI technique [Xu et al., 2010] which has been explained in page 92. The values are initiated to zero at the sink and they are updated at each node which receives. The receiver nodes need to record both the values to select energy efficient and/or minimum delay routes. The paths with fewer TC are more energy efficient as they are shorter in terms of Euclidean distance. On the other hand, the routes with fewer HC have lower communication delay as they are established by a fewer number of intermediate nodes. It gives CBA the ability to support low cost and/or delay communication links according to the data consumer and/or QoS requirements. When a cost message is received, the RRSI value of the link is measured and added then to TC value. Indeed, TC value gives the total RSSI weight of the route from the sink to the node. Besides, the receiver node increases the value of HC by one. The value of HC shows the node hop count from the sink. However, multiple cost messages are received through different routes during the cost value allocation phase. The messages have variant TC and HC values according to the link length and intermediate hop count. A cost message is immediately discarded if it has a greater HC than the node hop count. It is forwarded through a longer path (with a greater hop count) or a loop. The ID of sender node which has the minimum value of HC is recorded as TS (To the Sink) to establish a minimum delay Backward Path (BP). A BP is used to send the collected data from the event regions to the sink. If multiple messages with same HC values (with the node hop count) are received, the receiver node would record the ID of sender nodes as BackUp TS (BUTS) to establish alternative BPs if the primary one fails.

Each sensor node needs to wait for a while to receive a greater number of cost messages which is forwarded through other routes before re-transmitting the received cost message. It other words, sensor nodes miss some of the cost messages (which may have smaller TC) if they immediately re-transmit the received cost messages. Figure 4-2 depicts an example in which node D receives three cost messages through paths R1, R2 and R3. The network messages are normally received quicker through the paths with fewer hop count as the communication delay decreases. For this reason, it is assumed that the cost messages are received in the order of R1, R3 and R2. An inaccurate TC value is transmitted to the next hop nodes if D immediately transmits the cost message which is received through R1. In this case, TC(R3) should be propagated for the next

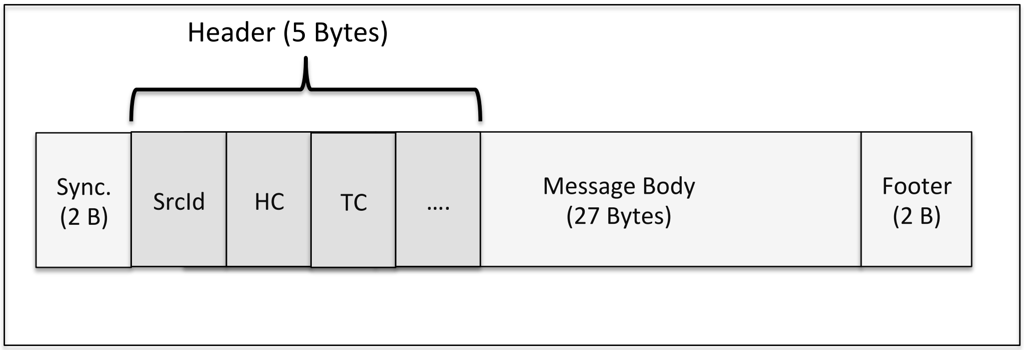


Figure 4-1: The structure of cost messages in CBA

hop nodes instead of TC(R1) as the latter is greater. This drawback is resolved similar to Minimum Cost Forwarding Algorithm (MCFA) [Ye et al., 2001]. It lets the nodes wait for a short period (back-off) until other cost messages which have smaller cost values are received. However, the waiting time increases network deployment delay. Moreover, it increases energy consumption as the nodes need to consume a greater amount of energy to receive the cost messages which are forwarded through different paths. For this reason, the back-off time should be minimised as much as possible. MCFA [Ye et al., 2001] recommends 10 ms for back-off time before sending the cost messages at each nodes.

4.3.3 Forming Data-Centric Clusters

CBA utilises clustering to establish an hierarchical routing infrastructure for data aggregation routing. With clustering, network congestion and energy consumption is reduced due to a decrease in the number of transmitted data packets [Liu, 2012]. As data packets are aggregated at the CHs, the source nodes do not need to forward their data to the sink individually. In consequence, clustering results in a reduction of transmitted network traffic, which decreases energy consumption and network congestion. CBA partitions the network into a set of clusters which are dynamically formed in a data-centric manner. Each cluster can be considered as a super node which has a particular data type to report/collect. DC clustering is used in CBA for the following reasons:

1. **Energy saving:** network resource consumption (mainly energy) is reduced due to three reasons: (1) Reducing the routing overhead: in DC clustering, nodes participate in clustering procedure if their data matches the data consumer interests. It results in reducing the amount of transmitted routing information (network traffic) especially in comparison to AC clustering in which any node

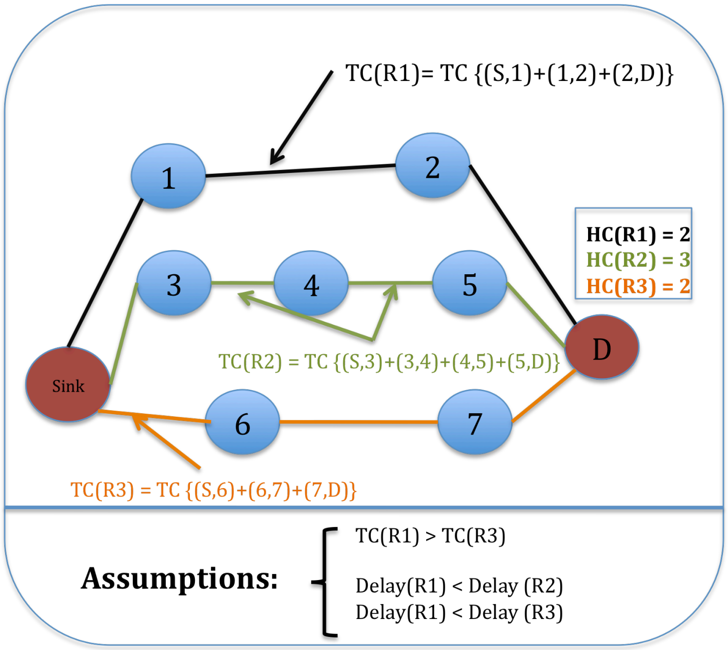


Figure 4-2: Cost value allocation in CBA

participates in clustering and/or route discovery/establishment. (2) Decreasing network overhearing: localising network transmissions into the clustered communications reduces overhearing that results in saving energy. (3) Reducing idle-listening: idle-listening reduces as the number of nodes which try to access wireless channels is reduced in DC clustering as compared to AC. In DC clustering, only the nodes which reside on DC inter/intra-cluster communications would need to access the wireless channels to forward the data packets. Hence, the number of nodes which stay available to communicate (or establish the routes) is reduced in comparison to address-centric clusters in which any node needs to frequently or periodically check the availability of wireless channels to establish an AC link.

2. **Enhancing accuracy:** DC clustering increases the accuracy of data aggregation by increasing data packet delivery. The network traffic reduces as the communications are limited into a set of DC clusters instead of the whole network (as in AC clusters). Reducing network traffic in DC clustered networks decreases network congestion and consequently increases data packet delivery. Hence, the accuracy of data collection is increased as a greater number of collected data samples is delivered for aggregation.

3. **Reducing delay:** data collection delay is reduced in DC clustered network as compared to Address-Centric (AC). AC clustering (e.g. LEACH) partitions the whole network into the clusters according to the address of nodes, whereas DC clusters are formed if the node has desirable data to report. For this reason, a smaller number of clusters is formed when a DC approach is used. It results in reducing the number of nodes (CHs) which intend to simultaneously access to wireless channels and consequently decreasing the transmitted network traffic in DC clustered networks as compared to AC. As a result, DC clustering reduces data collection delay due to decreasing the network traffic which results in reduction of the communication delays (wireless channel access and/or queueing).

4. **Supporting data collection in RS or ER event source distribution models:** DC clustering interconnects the random distributed source nodes as ER regions. It results in reduction of network traffic and routing overhead. Interconnecting the random scattered source nodes via data-centric links as DC clusters would logically forms integrated (ER) event regions in the network. Hence, data packets would get aggregated sooner and with a lower cost. This means that forwarding data packets from random scattered source nodes which are integrated via DC links reduces the number of transmissions in comparison to RS data collection. This is because it provides in-network data aggregation over DC links in comparison to AC links, in which each source node needs to individually establish a link to the sink or intermediate aggregators to transmit the data.

CBA utilises Hamming distance technique to form the clusters. The source nodes are grouped into a set of DC clusters according to the measured data type. Each cluster is managed by a single CH. The clustering procedure is started when the source nodes broadcast data advertisement messages (ADV_{msg}). The structure of ADV_{msg} is similar to hello messages, used by ZMA in Chapter 3 for vicinity discovery. The message header has four fields which are encoded with the Hamming technique: node data type, residual energy level (reliability), hop count (location) and TC (link cost). Residual levels of energy, hop count (HC) and link cost (TC) allow the nodes to select the most powerful nodes (in terms of having sufficient energy resources), which have minimum link cost to the sink. Data type is used to form data-centric clusters. The clustering procedure is performed in a distributed manner by the source nodes which are connected (in single-hop) and have the same Hamming code for their data types. When a ADV_{msg} message is received, the node updates its routing table by the sender ID, data type, energy level and HC. The nodes, which have the same Hamming code for data and HC, are grouped into the clusters. Then, each node considers its routing

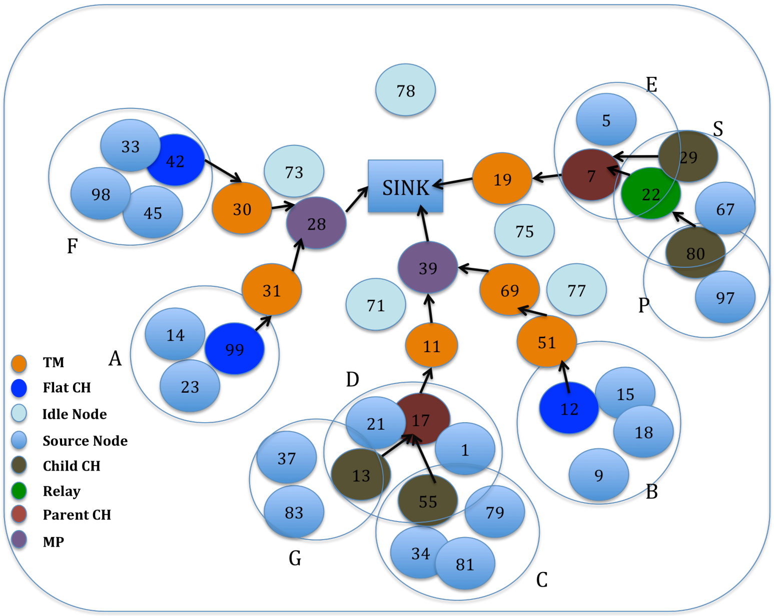


Figure 4-3: Clustering in CBA

table to find the strongest node which has the greatest level of energy. If there are multiple nodes which have same energy level, the node with smallest TC is selected as CH because it would be able to communicate with the sink via a minimum cost link if there is still a tie. The source nodes send (in unicast) a MY_{CH} message to the candidate CH to inform it about the selection. The node which receives the messages becomes the CH. Finally, the CMs are allocated by a cluster number according to their CH ID and data type.

CBA clusters are formed under two schemes: flat and hierarchical. A flat cluster is formed with no interconnection to any other clusters, whereas an hierarchal cluster is connected to at least one node which belongs to the next cluster. In the former, a CH does not know any other node in its single-hop neighbourhood which has a greater level of energy or a different cluster number. On the other hand, the CH has a connection to at least one node that belongs to another cluster in hierarchical clusters. The hierarchal clusters are interconnected through inter-cluster links which are used during the data aggregation routing to forward data packets. The cluster in the topmost level of hierarchy (with smallest HC) is called the parent cluster. According to Figure 4-3, for example, cluster A is flat as its CH (99) is not connected to any other cluster, whereas clusters P, S and E (parent) are hierarchical as they are interconnected.

Table 4.1: A Glossary for Tree Establishment Algorithm

Acronym	**Definition**	**Acronym**	**Definition**
RREQ	Route REQuest packet	HC	Hop Count
RREP	Route REPly packet	TC	Total Cost
srcid	the last sender ID	BP	Backward Path
FP	Forward Path	BHC	BP Hop Count
BTC	BP Total Cost	TS	To the Sink
BUTS	BackUp TS	TM	Tree Member
FRR	Failed Route Request	MP	Meet Point
APC	Aggregation Path hop Count	CH	Cluster Head

4.3.4 Data Aggregation Routing

To reduce the cost of data reporting, CBA forms a tree-based backbone from the CHs to the sink to collect and aggregate data via minimum hop-count links. The tree infrastructure is established to aggregate and forward data packets from the event regions to the sink in an hierarchical manner. It establishes minimum hop count paths from the event regions to the sink, in which data packets get aggregated as soon as possible. Hence, forwarding data packets through the tree infrastructure leads to a reduction in the number of transmitted data packets and consequently the data reporting cost. This section describes the tree establishment algorithm. Table 4.1 decodes the acronyms which are used in data aggregation routing algorithm.

The tree infrastructure is established by interconnecting the minimum hop count routes from the CHs to the sink. It is rooted in the sink and formed in a data-centric manner using route request packets (RREQs) and route reply packets (RREPs) which are forwarded between the CHs and sink. The RREQ messages are forwarded from the CHs to establish inter-cluster routes and/or interconnect the data regions (DC CHs) to the sink, whereas RREP messages are forwarded from the sink to the CHs to form the tree branches. Establishing the tree backbone in a data-centric manner supports hierarchical data aggregation in CBA from the event regions to the sink. In addition, it allows CBA to work over heterogeneous WSNs in which the nodes have variant data types to report. This means that CBA is able to collect interesting data samples from the source nodes with variant data type as the tree infrastructure is dynamically formed from DC CHs to the sink based on the data consumer interests.

To form the tree backbone, CBA utilises a parallel collision-guided technique. It is conceptually similar to [Basurra, 2012], but differs in implementation and execution. The parallel collision-guided approach has the potential to reduce the overhead of the backbone establishment. It avoids to forward redundant and/or useless route requests (RREQs) through the links which are already established. The RREQ messages are

```
Data: Routing Table (RT)
RREQ initialisation at CH:
if Node Role = (Flat or Parent CH) then
    $RREQ \leftarrow (D, CID, srcId, BHC = 0, BTC = 0)$;
    if TS is not available then $TS \leftarrow Select(BUTS)$;
    $RREQ \rightarrow TS$;
end
RREQ forwarding at intermediate nodes (Node X):
/* Update RT                                                    */
if $RREQ.CID \in RT$ then
    if $RREQ.BHC < RT.BHC$ then
        /* RT is updated                                        */
        $RT(CID) \leftarrow (CID, RREQ.srcId, RREQ.BHC, RREQ.BTC)$;
    else
        Discard (RREQ);
    end
else
    Insert RT(CID, srcId, BHC, BTC);
end
/* Forward RREQ                                                 */
if Role = (Sink) then
    Delete RREQ;
else if (Role = CH or CM) and $(X.CID \neq RREQ.CID)$ then
    $X \leftarrow Parent(CID)$;
    Delete RREQ;
else
    if Count (RT) > 1 then
        /* RREQ collision                                       */
        Delete RREQ;
    else
        $srcId \leftarrow X.Id$;
        $BHC \leftarrow BHC + 1$ ;
        $BTC \leftarrow BTC + RSSI(link)$;
        $RREQ \leftarrow (D, CID, srcId, BHC, BTC)$;
        $RREQ \rightarrow Select(TS \parallel BUTS)$;
    end
end
```

Algorithm 3: RREQ Forwarding Algorithm

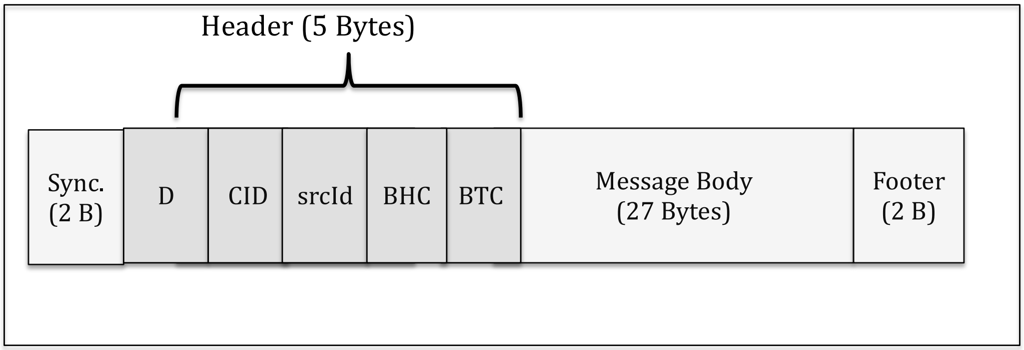


Figure 4-4: The structure of RREQ in CBA

forwarded in parallel from each data region which is formed as flat and/or hierarchical clusters to the sink. In hierarchical clusters, RREQs are forwarded only by the parent CHs. This is because the child clusters are hierarchically interconnected to the parent cluster(s) during the clustering procedure. Hence, establishing a path from a parent cluster to the sink results in interconnecting all its child clusters to the sink. This reduces the number of transmitted RREQs and routing overhead in hierarchical clusters. For example, cluster-head 17 is the only node that forwards a RREQ to the sink from clusters C, D and G which are hierarchically interconnected according to Figure 4-3.

As Figure 4-4 shows, the header of a route request packet (RREQ) has five fields: cluster data type (D), CH id (CID), last sender ID (srcId), BP Hop Count (BHC) and BP Total Cost (RSSI) value (BTC). The values of BHC and BTC are measured as the path hop count and RRSI on Backward Paths from the node to the sink. Although hop count and path RSSI value are measured on FPs using HC and TC values, they are calculated similarly on BPs using BHC and BTC to inform the sink about the location of DC CHs and/or select the tree paths to forward the route reply packet (RREP) through. Besides, BTC and BHC can be used to support uni-directional communications when RREP messages (to establish a BP) are forwarded via the same FP from the CH to the sink. Wireless communication routes are formed unidirectional due to variant communication power and signal propagation (especially in heterogeneous networks), ambient noises and/or wireless communication fading [Ramasubramanian et al., 2002]. This is stated formally in Algorithm 4-4.

The RREQ messages are forwarded from flat and/or parent CHs to the sink. According to Algorithm 4-4, the CHs firstly check the availability of their TS nodes to forward the RREQ messages. The messages are forwarded if the TS nodes are available. Otherwise, a BUTS node (or neighbour node which has a smaller HC) is selected by each CH to forward the RREQ message. It reduces the overhead of forwarding

```
Data: RREP Table (PT))
RREP initialisation at Sink:
if Role = (Sink) then
    $RREP \leftarrow (CID, srcId, APC = 0, Query)$;
    $RREP(s) \rightarrow srcId(s)$;
end
RREP forwarding at intermediate nodes (Node X):
if $CID \in PT$ then
    Discard (RREP);/* redundant RREP */
end
Insert PT(CID, srcId, APC);
if Role = (CH or CM) then
    if X has a child (CID) then
        $RREP \leftarrow (CID, srcId, APC = 0, Query)$;
        $RREP(s) \rightarrow CID(s)$;
    else
        Delete RREP;/* a tree branch is established */
    end
else
    if Count (RREQ) $\not>$ 1 then
        /* checks the number of received RREQ messages */
        $TM \leftarrow X$;
        $RREP \leftarrow (CID, srcId, APC++, Query)$;
        $RREP \rightarrow srcId$;
    else
        $MP \leftarrow X$;
        /* for the CH whose RREQ is received by the sink */
        $RREP \leftarrow (CID, srcId, APC++, Query)$;
        $RREP \rightarrow srcId$;
        /* for the CHs whose RREQ collided at MPs */
        $RREP \leftarrow (CID, srcId, APC = 0, Query)$;
        $RREP(s) \rightarrow srcId(s)$;
    end
end
```

Algorithm 4: RREP Forwarding Algorithm

RREQ as compared to message broadcasting. Each intermediate node, which receives a RREQ, would record srcId value of the message as a potential link to a desirable CH. The receiver node updates then the RREQ by changing the srcId to its id, increment BHC by one and add the last link RSSI value to the BTC. RREQ messages are forwarded in a same manner until the following situations arise:

1. The RREQ is received by the sink. In this case, the srcId value of the received RREQ would show a single-hop neighbour which knows a potential path with BHC hop count and BTC cost to a desirable CH.

2. The RREQ collides with another at an intermediate node. In this case, the intermediate node stops forwarding the RREQ messages(s) to the sink to conserve network resources.

3. The RREQ is received by CM/CH which has lower HC value. The cluster which receives the RREQ becomes a parent of the sender cluster.

Route reply packets (RREP) are forwarded from the sink through the shortest paths which have the minimum BHC/BTC to the desirable CHs (in terms of data). RREP can include data collection requirements, thresholds and attributes similar to TEEN [Manjeshwar and Agrawal, 2001] and APTEEN [Manjeshwar and Agrawal, 2002]. The header of RREP maintains the message sequence number, CID (the target cluster id), srcId (the next hop neighbour node) and APC (Aggregation Path hop Count). APC will be used during data aggregation routing to find the path distance (hop count) from the CH to the sink or the (closest)intermediate node which has the potential to aggregate data. The value of APC is initially zero and incremented at each intermediate node until the following situation arises (see Algorithm 4):

1. The RREP is received by a cluster (CM or CH). In this case, the cluster avoids to forward the RREP unless it is connected to any other cluster with higher HC value (child cluster). A copy of the received RREP is forwarded to each child cluster if there is still a tie. The value of APC is set to zero in copy RREPs. The copy RREPs are forwarded until they are received by the target cluster.

2. The RREP is received by a node which already has received multiple RREQs. The node becomes an MP and the original RREP is forwarded to the target cluster. In addition, a copy of RREPs (APC=0) is forward to each cluster which is connected via a single or multi-hop link.

3. The RREP is received by a node which already has received a single RREQ. The node becomes an TM and the RREQ is forwarded to the destined cluster.

Data aggregation is started in a bottom-up manner from the child and/or flat clusters. The data samples are locally aggregated at the CHs. Then, the aggregated results are forward from the CHs to the parent clusters or sink through intermediate TMs. Each CH selects the path with the minimum APC (aggregation path) to forward its data packets through if multiple RREP messages are received. This means that CBA supports early data aggregation in which data packet get aggregated at the closest aggregator on the tree as soon as possible (with minimum hop count). The data results from each cluster are aggregated at MPs or parent CHs and then hierarchically forwarded until finally received by the sink. It forms a minimum hop count tree in CBA that has the potential to minimise end-to-end delay. The tree branches provide the minimum hop count links between the CHs, TMs and MPs in which data packets are aggregated with minimum delay. According to Figure 4-3, for example, aggregated results from cluster G, C are combined at D. Then, CH (17) transmits the aggregated result to MP 39 to be combined by the result of cluster B. MP 39 is then responsible for forwarding the aggregated result of cluster G, C, D and B to the sink.

4.4 Experimental Plan

To test and evaluate CBA, we use simulation technique. OMNET++ [OMNET++, 2012], is an open-source simulator that was used to simulate LEACH [Heinzelman et al., 2000] and Directed diffusion [Intanagonwiwat et al., 2000]. It has a modelling framework called MiXiM [Viklund, 2013] that offers detailed models of radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols such as B-Mac in WSN.

The experiments are designed for client/server routing according to three levels of parameters similar to Chapter 3: area size, node count and data density (the proportion of interesting data). The experiment parameters (similar to Figure 3-7 in Chapter 3) are designed to observe the protocol behaviour, scalability and performance based on varying area size (small, medium and large), node count (sparse and/or dense), and the proportion of interesting data samples in the network (25 to 100 percent). The total number of delivered data samples, average end-to-end delay, total hop count, total transmitted traffic[1] and total consumed energy are the metrics which are selected to be measured in the experiments. Measuring energy consumption, end-to-end delay and accuracy is required in evaluating data aggregation routing protocols due to the energy resource constraints in WSN and data consumer requirements (deliver maximum data samples with minimum delay). Path hop count and transmitted network traffic also

---

[1] As Chapter 3, this term refers to the total sent and received network traffic

are measured as they have a high impact on energy consumption, accuracy and/or end-to-end delay. In other words, investigating about path hop count and network traffic would show the efficiency of routing algorithms to save the energy, reduce delay and increase accuracy.

4.4.1 Simulation Setup

All the experiments in this chapter are set up similarly to those in Chapter 3. They are implemented using MiXiM [Viklund, 2013] and run 300 times, each for 3600s, to perform a reasonable number of data aggregation routing attempts and have sufficient time to complete the process. A single sink resides in centre of the sensing field to collect the report data samples. The sink has a sensor attached to sense/measure the ambient events in all the experiments. The sensor nodes are modelled as a set of static wireless nodes that are randomly distributed in the network field. They are powered by limited lifetime batteries that are 99999 mAh and 3.3V. A Line-Of-Sight (LOS) model [Uthansakul et al., 2005] is used for wireless signal propagation and the nodes radio range is 75 metres (constant for all nodes). Appendix B.2, shows the simulator experiment setup file for CBA.

Similar to Chapter 3, three parameters are considered to test and evaluate the performance of CBA: area size, node count and data density. As Figure 3-7 shows, the experiments are carried out for three different size of areas: small (200×200 m^2), medium (400×400 m^2) and large (800×800 m^2). The different area sizes let us to observe the protocol behaviour and performance when the area size is small and the communications can be formed in single-hop between the nodes, or the deployment area is large and the nodes need multi-hop links to communicate.

A network is deployed with three different node count at each area size to test the protocols scalability. Varying the node count lets us observe the protocol behaviour, scalability and performance when the network deployed is sparse or dense. According to Figure 3-7, three different network sizes are deployed in each size of area: small network (200×200) with 16, 32 and 64, medium (400×400) with 64, 128, 256 and large (800×800) with 256, 512, 1024 nodes respectively. Each network is deployed in four scenarios according to the proportion of interesting data samples (data density) to the total node count in the area. This means that each network is allocated with four different amount of data samples (25%, 50%, 75% and 100%) which need to be collected/reported. The experiments allow us to observe ability and performance of the protocols to route the data packets to the sink according to the varying proportion of data samples in the network. Moreover, it exhibits how the protocols deal with the network traffic and message conflicts when the number of data transmissions is increased.

4.4.2 Performance Parameters

Various quantitative metrics are used to analyse the performance of the routing protocols. The parameters chosen to be measured in this chapter are: total consumed energy, total number of delivered data samples (accuracy), average end-to-end delay, average hop count and total transmitted traffic. They are explained briefly in turn as below:

1. **Total consumed energy:** represents the total amount of energy that is consumed to establish, deploy and maintain the routing infrastructure and/or route the data packets to the sink. Minimising energy consumption leads to maximising the network lifetime. Reducing the overhead of routing infrastructure deployment/maintenance and communication overhead are two key factors that have the potential to reduce the energy consumption.

2. **Total delivered data samples (accuracy):** calculates the number of data samples collected by the sink through direct or indirect links. This metric has a (positive)correlation with data aggregation robustness. This means that maximising the number of data samples in data aggregation procedure would result in enhancing the accuracy of the summary results for the data consumer. Hence, the consumer would be able to make better decisions on the collected data. However, data transmissions fail due to message collisions, network traffic and/or lack of routing capability. For this reason, accuracy is measured to show the ability of the routing protocols to reduce the message collision/traffic and establish efficient and reliable paths.

3. **Average hop count:** measures the capability of routing protocols in establishing optimal/shortest paths to forward data samples. It is calculated as the average intermediate hop count from the event regions to the sink for each delivered data sample. The total hop count of paths (which are established for all reported date packets over inter or intra cluster links) is divided by the number of collected data samples at the sink to calculate the average hop count. The average hop count influences the network energy consumption and data aggregation delay. The network resource consumption and data collection delay would reduce if data packets are delivered to the sink through minimum hop count links. On the other hand, establishing random/blind paths increases the number of intermediate hops on routes resulting in increased communication cost and delays. For this reason,

efficient routing protocols aim to reduce hop count which reduces data collection delay and saves network resources.

4. **Average end-to-end delay:** this measure the average end-to-end delay (ETE) of data aggregation routing. ETE is measured from when the data samples leaves the source nodes until they are collected and aggregated at the sink. It is influenced by communication and computation delays such data packet reception and transmission, routing and aggregation latency. ETE has the potential to influence data accuracy and freshness. This means that data packets are expired or lose their meaning if they are delivered to the sink late. For this reason, the objective of routing protocols is to minimise the end-to-end delay (ETE) as much as possible.

5. **Total transmitted traffic:** represents the amount of network traffic transmitted (sent/received), including control and data packets, in entire network during data aggregation routing. Control packets are transmitted to deploy the network, establish/maintain the routing infrastructure and/or route the data packets, whereas data packets are forwarded to deliver the measured data samples to the sink. Routing control packets include: Hello, route request/reply, route errors and maintenance, routing updates and acknowledgements. Increasing the network traffic would result in increasing the network resource consumption. Besides, end-to-end delay (ETE) is increased due to increasing wireless channel access and communication delays. Hence, reducing the transmitted network traffic results in reduction of energy consumption and data aggregation end-to-end delay.

4.5 Results

This section evaluates the performance of CBA, LEACH [Heinzelman et al., 2000] and Directed diffusion [Intanagonwiwat et al., 2000] based on the routing performance metrics that are described in the previous section. Each metric is measured in 36 scenarios according to area size (three), node count (three) and proportions of interesting data sample in the network (four).

4.5.1 Total Energy Consumption

Introduction: Energy saving is a vital requirement in WSN because of the sensor node power resource constraints. The objective of designing WSN protocols is to enhance the network lifetime by reducing power consumption. This section evaluates the energy efficiency of CBA in comparison to two benchmarks namely LEACH and DDiFF.

Evaluation: As Figure 4-5 shows, CBA consumes a lower amount of energy to collect and aggregate data samples over a clustered network in comparison to LEACH. This is for the following reasons:

1. Utilises the Hamming distance technique to form the clusters: CBA uses Hamming distance, which is a lightweight technique to from clusters in WSNs. The number/amount of transmitted information to form the clusters is reduced as compared to LEACH, that needs periodically to collect routing information to form the clusters and select the cluster-heads. The clustering cost and routing information collection is increased in LEACH, especially when the node count increases due to increasing overhearing and/or the number of node which participate in the data clustering procedure.
2. Forming DC clusters: this reduces the network energy consumption (especially when the proportion of interesting source nodes is low) because the nodes participate in the clustering procedure if they have interesting data to report. However, all nodes need to participate in clustering when an address-centric clustering approach such as LEACH is utilised to form the clusters.
3. Avoids heuristic packet routing: CBA utilises a tree backbone to forward aggregated results from the clusters to the sink. It results in a reduction of the cost of transmitting the data packets from the cluster-heads to the sink. The data packets are forwarded through shortest paths (spanning tree), which are established from the data regions to the sink. They are hierarchically aggregated through the tree infrastructure until the final result is received by the sink. Hence, the number of messages and communication hop count are reduced as compared to LEACH, in which the paths are heuristically established from the data regions to the sink to forward data packets.
4. Hierarchical aggregation over tree infrastructure: data packets in CBA are hierarchically aggregated and forwarded through a tree infrastructure from the event regions to the sink. It results in a reduction of the number of data packet broadcasts, as the TMs (tree members) utilise unicast to forward the data to the sink. However, lack of such infrastructure in LEACH increases the number of data transmissions as a greater number of intermediate/CH

nodes needs to broadcast/multicast the data packets, to heuristically route them to the sink.

5. Utilises parallel guided-collision: CBA utilises parallel guided-collision to reduce the overhead of establishing the tree infrastructure. This technique avoids forwarding/broadcasting useless/redundant control packets during the data aggregation infrastructure establishment phase. It results in reducing the number of control packets and consequently decreasing the total transmitted network traffic.

Energy consumption in CBA is increased, compared to DDiFF when data density increases in the network. DDiFF does not establish a hierarchical routing/aggregation infrastructure for data aggregation. Hence, it consumes a lower amount of energy in comparison to CBA which forms the clusters and creates the data aggregation tree to collect and aggregate data samples. Energy consumption in CBA is increased when the number of desirable source nodes which participate in DC clustering and tree establishment phases increases.

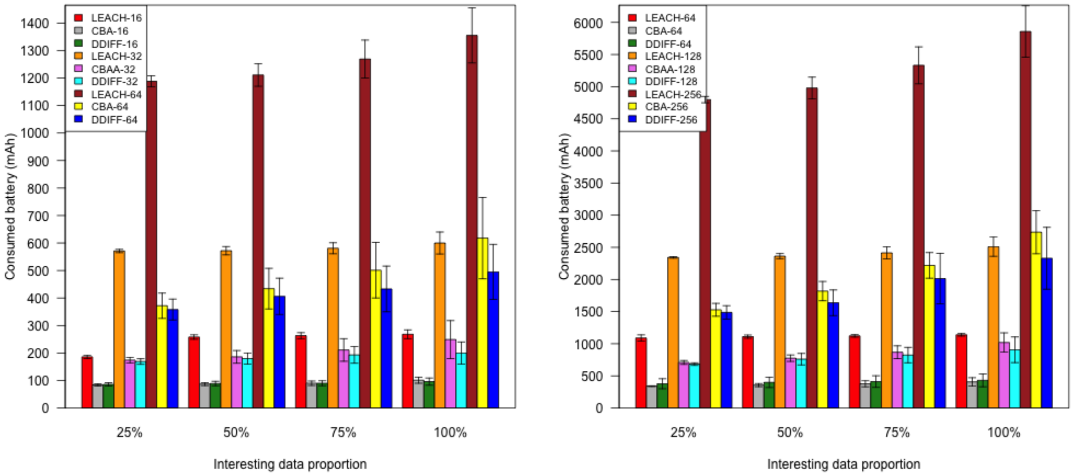
4.5.2 Total Number of Captured Data Samples (Accuracy)

Introduction: Accuracy is measured as the number of interesting data samples that are collected at the sink. Data samples are reported and aggregated through flat or hierarchical infrastructures until they are received by the sink. The objective is to maximise the accuracy as it has the potential to enhance data aggregation robustness. This section evaluates the accuracy of CBA as compared to LEACH and DDiFF.

Evaluation: From Figure 4-6, it is observed that CBA outperforms DDiFF and LEACH when node count and/or data density (the proportion of desirable nodes) increases in the network. It is rooted in three reasons as below:

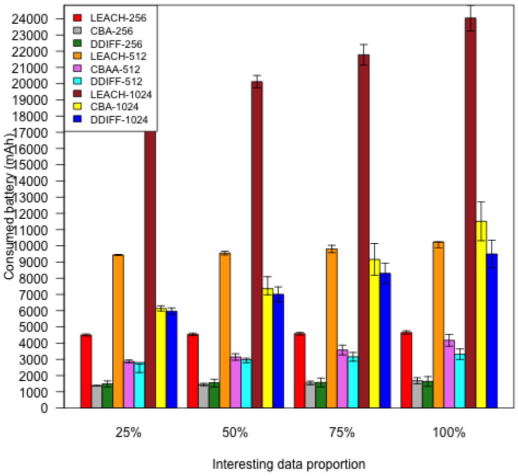
1. Partitions the network into the clusters: clustering has the potential to reduce the message collision as data transmissions would be limited into inter/intra-cluster communications. Using the cluster-based infrastructure in CBA, data packets are forwarded from the source nodes to CHs through intra-cluster DC links. The CHs collect and aggregate intra-cluster data samples and then report the aggregated results to the sink. For this reason, fewer nodes (CHs) need to access wireless channels to transmit their data as compared to flat networks (i.e DDiFF) in which any source node individually forwards its data. This reduces the probability of message collision caused

Figure 4-5: Energy consumption of client/server routing protocols.



(a) small area (200×200)m^2

(b) medium area (400×400)m^2



(c) large area (800×800)m^2

Key points of the figures:

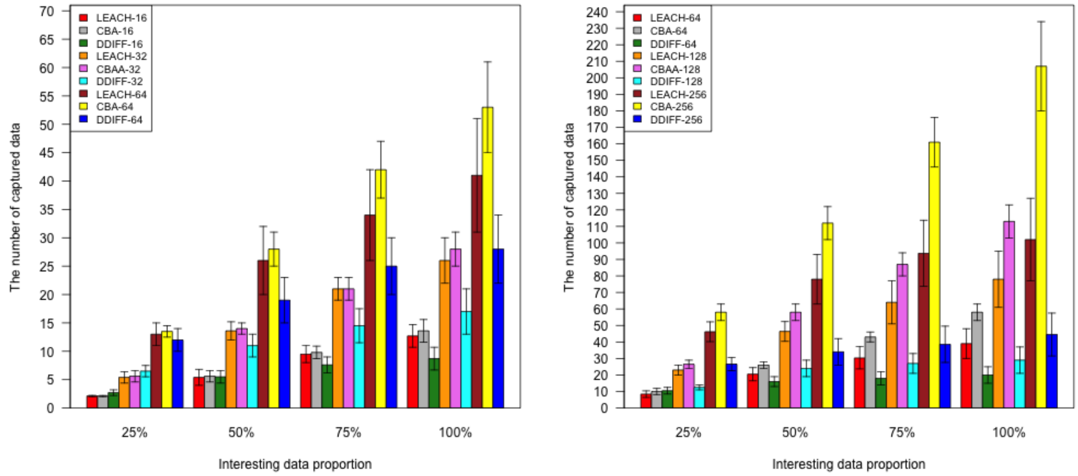
1. CBA outperforms LEACH in terms of energy consumption.
2. Energy consumption of LEACH is significantly increased when the deployed network is dense.
3. DDiFF outperforms CBA especially when the data density increases.

by communication invention especially when data density increases in the network. Owing to this, the accuracy of CBA is enhanced (in comparison to DDiFF) when node count and/or the proportion of desirable source nodes increases.

2. Forms the clusters in DC manner: clustering the network in a DC manner has the potential to reduce message collisions in comparison to AC clustered network. CBA partitions the network in a DC manner. It reduces cluster count and consequently the CHs which try to access wireless channels to forward data packets to the sink especially when data density is low in the network. On the other hand, the probability of message collision is increased in LEACH as the number of CHs is increased when whole the network is portioned into AC clusters without considering the data content at the source nodes.
3. Utilises the aggregation tree to forward data packets from the event regions to the sink: the CHs utilise a reliable infrastructure in CBA that is established to collect, aggregate and forward data samples from the source nodes to the sink. It reduces the probability of message conflict/loss in comparison to LEACH which heuristically route the data packets from the source nodes to the sink. Lack of a reliable infrastructure from the event regions to the sink to forward the data packets increases the probability of message conflict/loss especially when node count and/or data density is high in the network. This means that data packets is lost/collided when a great number of CHs dynamically route the packets from the event regions to the sink.

The accuracy of CBA increases when the network size rises. Utilising the tree backbone to forward the data packets from the source nodes to the sink, is the key reason for the increase in accuracy in CBA. In large networks, the probability of message conflict/loss is increased in routing protocols which heuristically route data packets (i.e LEACH) as the hop count between the source nodes and sink is increased. Increasing the hop count between the event regions and sink increases the number of intermediate nodes which participate in routing data packets. Hence, data messages have a higher chance to fail due to collision as a greater number of nodes try to access the wireless channels to forward them. Moreover, the probability of message loss goes up with the number of relay nodes – these heuristically route the data packets from the event regions to the sink. On the other hand, CBA utilises a tree backbone in which data packets are forwarded through reliable and/or shortest path to the sink. The probability of

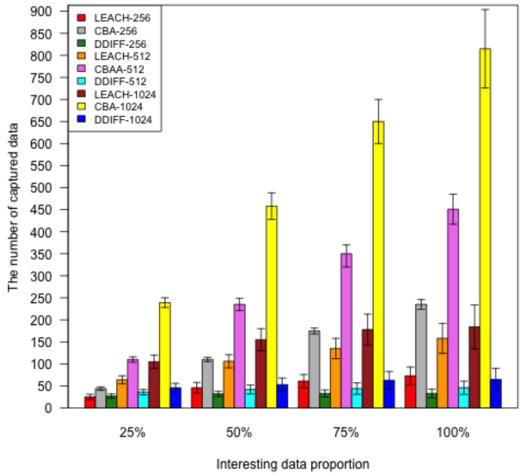
Figure 4-6: Accuracy of client/server routing protocols.



(a) small area (200×200)m^2 (b) medium area (400×400)m^2

Key points of the figures:

1. CBA outperforms the benchmark protocols in terms of accuracy.
2. LEACH has a better performance in dense network as compared to DDiFF.
3. CBA increases accuracy when network deployed is dense and large.



(c) large area (800×800)m^2

message collisions/loss is reduced as the number of relay nodes would be limited to the tree members (TMs). Each node which resides in the tree utilises unicast communication instead of broadcast/multicast to collect the data packets from child nodes and then forwards the aggregated result to the parent nodes. Hence, routing the data packets via the tree infrastructure would result in reduction of communications and consequently message collisions/loss.

4.5.3 Average Hop Count

Introduction: Average hop count is measured as the average path length from the event regions to the sink to report each data sample. The objective of a routing protocol is to reduce the average hop count as much as possible. The average hop count influences ETE and energy consumption. This means that increasing the average hop count results in increasing the number of intermediate nodes which participate in sending/receiving the data packets. Hence, energy consumption and the average end-to-end delay increases. Energy consumption increases because of increasing the number of intermediate nodes which route the data packets from the event regions to the sink. Moreover, ETE is increased due to increasing communication delays over longer paths which comprise of a greater number of nodes. For this reason, data aggregation routing protocols need to minimise hop count by establishing shortest paths in which data packets are aggregated as soon as possible.

Evaluation: According to Figure 4-7, CBA reduces the average hop count in comparison to LEACH and DDiFF. Data-centric clustering and utilising the spanning tree for data aggregation routing are two techniques that reduce the average hop count in CBA. DC clustering and spanning tree would reduce path hop count as below:

1. DC clustering: CBA forms DC clusters to collect and aggregate local data samples according to the sink interests. Data samples are forwarded via intra-cluster links to the CHs for aggregation. Thus, data packets do not need to traverse long paths until they are heuristically aggregated at intermediate nodes or sink as they are aggregated at CHs. On the other hand, data aggregation is not guaranteed in address-centric CHs as AC clusters (intra-cluster links) are formed based on the address of nodes and not the content of data. In AC clusters (i.e LEACH), a node performs data aggregation if it resides on a route through which multiple data packets are forwarded. Thus, data packets need to traverse longer paths (especially

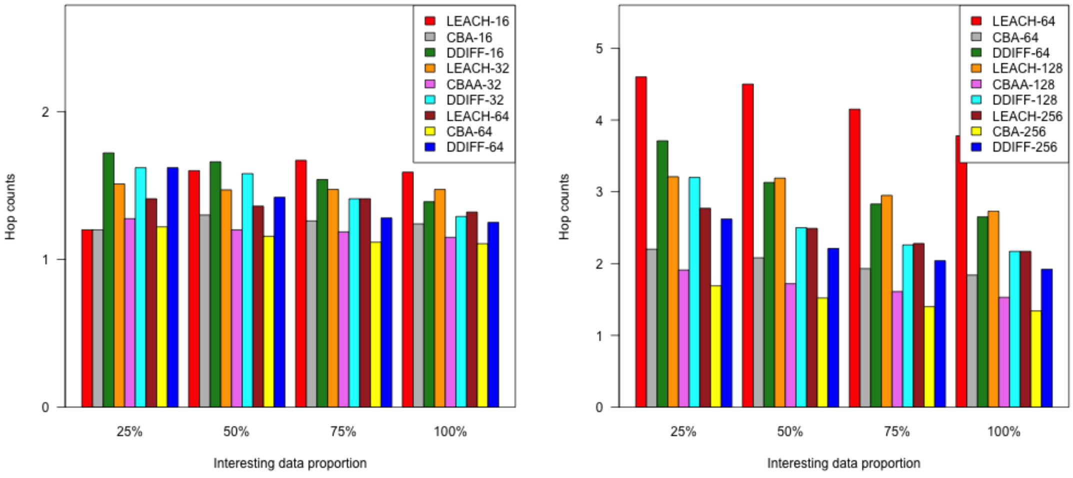
when data density is low) until get aggregated at AC aggregator nodes in comparison to DC clusters in which data aggregation is performed at each CH. For this reason, the average hop count of LEACH is higher compared to CBA.

2. Utilises a tree backbone: the aggregated result of each cluster is forwarded to the sink through a spanning tree which is formed by the shortest paths (using HC value) between the event regions and the sink. Data packets are hierarchically forwarded and aggregated until the final result is received by the sink. Hence, the data packets would traverse through minimum hop count paths from the event regions to the sink. On the other hand, the data packets are heuristically forwarded from the event regions to the sink in LEACH. It increases the path hop count in LEACH in comparison to CBA especially when the area size and the number of intermediate nodes increases. Moreover, the lack of a hierarchical infrastructure in DDiFF to aggregate data packets is the reason for the increase in average hop count. This means that data packets need to traverse longer paths until they are get heuristically aggregated at the intermediate aggregators or sink. As a result, the average hop count in DDiFF is higher compared to CBA, in which data packets are forwarded and aggregated through a clustered network.

DDiFF reduces the average hop count as compared to LEACH especially when the network deployed is sparse. This behaviour derives from DDiFF forwarding the data packets through shortest paths (which are established during route establishment phase) from the source nodes to the intermediate aggregators and/or sink for aggregation. On the other hand, LEACH heuristically establishes AC paths from the CHs to the sink to forward the data packets. As the network deployed is sparse, data packets have a lower chance to be aggregated at the address-centric CHs. They need to traverse through long AC paths heuristically from the event regions until they are received by the sink or are aggregated at the intermediate nodes which reside on joint (inter-cluster) paths. Hence, the average hop count is higher in LEACH as compared to DDiFF when the network is sparse.

Conversely, LEACH reduces the average hop count in comparison to DDiFF when the network deployed is dense and large. According to Figure 4-7, increasing the network interconnectivity (when the node count and data density increases) generally results in a reduction of the average hop count in the routing protocols. This is because dense networks increase the chance for data packets to be aggregated. In LEACH, data packets are more likely to be aggregated at CHs

Figure 4-7: Average hop count of client/server routing protocols.

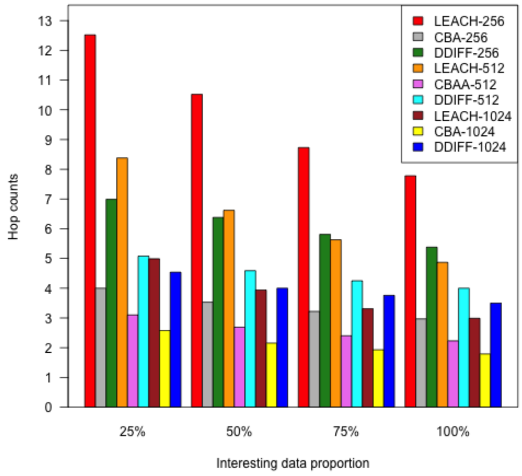


(a) small area $(200{\times}200)m^2$

(b) medium area $(400{\times}400)m^2$

Key points of the figures:

1. The average hop count is reduced when the node count increases.
2. CBA forwards data packets through shorter paths as compared to the benchmark protocols.
3. The average hop count in LEACH is increased if area size and/or node count increases.
4. DDiFF increases hop count in large and dense network as compared to LEACH.



(c) large area $(800{\times}800)m^2$

due to the increased the number of desirable source nodes in the network (or each cluster) when data density increases. This means that a greater number of data packets are aggregated at the CHs due to the increased the number of CMs which forward desirable data packets at each AC cluster when data density is increased. On the other hand, there is no hierarchical infrastructure to aggregate the data packets in DDiFF. Source nodes need to forward data packets individually through the shortest paths to the sink and/or intermediate aggregators. Owing to this, data packets are aggregated late as compared to LEACH in which data packets are aggregated at CHs when the network deployed is dense. Owing to this, the average hop count is increased in DDiFF as compared to LEACH.

4.5.4 Average End-to-End Delay

Introduction: Average end-to-end delay (ETE) is measured as the average time since data packets leave the source nodes until they are received by the sink. The objective of data aggregation routing protocols is to reduce the average delay as it enhances data freshness. Reporting the data samples to the sink as quick as possible provides the data consumer with fresh data for further analysis. Hence, the data consumer is able to make better decisions according to the collected data. There are two issues which have the potential to influence ETE in client/server data aggregation routings:

1. The routing scheme: it can be either proactive or reactive to find/establish paths and forward data packets. Proactive routing protocols usually reduce ETE as the paths are already established to forward the data packets from the event regions to the sink. Hence, the sensor nodes do not need to collect routing information to discover and establish the paths. On the other hand, ETE is increased in reactive routing as the sensor nodes need to collect required routing information to route the data packets from the data regions to the sink.
2. The communication delay: wireless communication delay has the potential to influence ETE in data aggregation routing. There are a set of parameters such as the distance between the sender and receiver, the number of intermediate hops, network traffic, message size and node characteristics (i.e operating system and/or queue size/delay) that influences the communication delays [Ardakani et al., 2014]. The communication delay is decreased by minimising the number of intermediate hops on the data collection paths. This means that the end-to-end delay can reduce when the data packets are

forwarded through minimum hop count routes.

Evaluation: As Figure 4-8 shows, CBA reduces ETE in comparison to DDiFF and LEACH especially when the area size and/or node count increases. It is because of three reasons as below:

1. Establishing shortest paths to report the data packets: CBA forwards data packets through a spanning tree which is formed by the shortest paths from the event regions to the sink. The data packets traverse through minimum hop count routes until received by the sink. The communication delay is reduced as a minimum number of intermediate nodes participate in routing data packets. On the other hand, LEACH heuristically routes the data packets from the CHs to the sink. It increases the path hop count (according to Figure 4-7) which results in increasing ETE in LEACH especially when the network deployed is large and dense.
2. Hybrid routing: CBA utilises a hybrid routing scheme in which data packets are reactively forwarded through intra-cluster links, whereas they are proactively routed via the tree backbone. Each CH collects data samples from its CMs and then forwards the aggregated result to the sink through the spanning tree. The TMs (tree member nodes) do not need to reactively collect routing information to route the data packets as they already know the required information. Hence, end-to-end delay is reduced as compared to LEACH in which CHs reactively route the data packets from the event regions to the sink.
3. Early data aggregation[2]: data packets are aggregated in CBA as early as possible at the CHs and/or TMs. It results in reduction of ETE as compared to the routing protocols in which data packets get aggregated late. Early data aggregation has the potential to reduce the number of data transmissions and/or relay nodes. This means that a fewer number of data packets needs to be forwarded when they are aggregated as early as possible (in terms of hop count). Moreover, the number of relay nodes which are in charge of packets forwarding from the event regions to the sink is reduced if the number of transmitted data packets is decreased. It results in a reduction of network node requests to access the wireless channels and consequently decreased idle-listening and/or access time. However, lack of early data aggregation increases the access time and consequently enhance ETE in the routing protocols such as DDiFF. Forwarding data packets by each source

---

[2] this term refers to aggregating data packets as soon as possible in terms of traversed hop count

nodes to the sink increases the number of sensor nodes which try to access the wireless channels in DDiFF. It results in increased access time delay and consequently increased ETE as compared to CBA as node count and/or area size increases.

From Figure 4-8, it is apparent that ETE is increased in LEACH as compared to DDiFF when the node count and/or area size increases. LEACH forwards the aggregated data packets from each cluster in a reactive manner to the sink. Hence, ETE is increased when the number of intermediate nodes increases between the CHs and the sink. On the other hand, reducing the path hop count and utilising proactive routing are two reasons for the reduction of ETE in DDiFF as compared to LEACH when the network is large and dense. DDiFF proactively forwards the data packets through shortest paths. It results in a reduction of routing and communication delay that consequently reduces ETE.

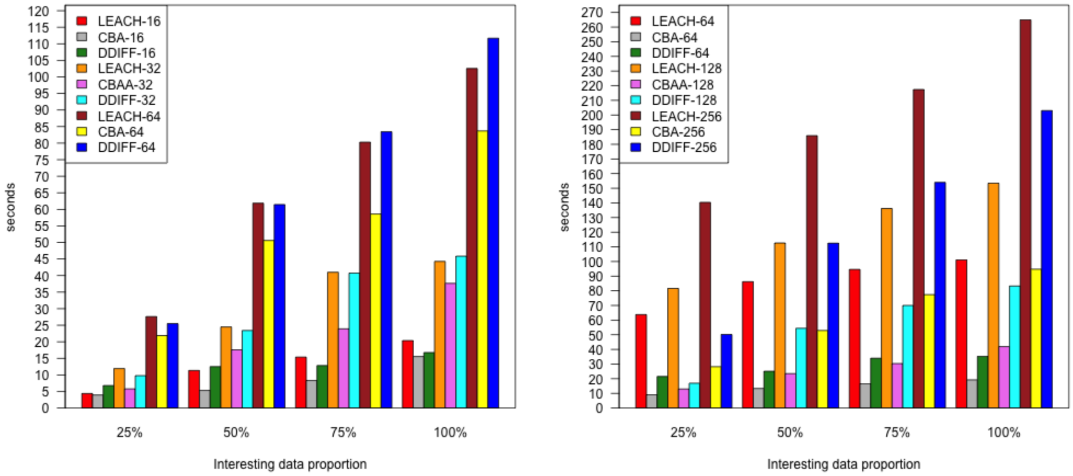
Conversely, ETE is reduced in LEACH as compared to DDiFF when a dense network is deployed in a small area. In LEACH, the CHs mostly are able to communicate with the sink in single-hop when the area size is small. Hence, the data packets do not need to be forwarded via heuristic routes, but they are delivered directly to the sink from the CHs. Moreover, early data aggregation at CHs reduces access time delay in LEACH when the network deployed is dense.

4.5.5 Total Transmitted Traffic

Introduction: Similar to Section 3.4, total transmitted traffic is measured as the total amount of data/control packets (sent/received) transmitted by the network nodes. Data packets are used to report the ambient data, whereas the control packets are transmitted to deploy the network, form the routing infrastructure and/or control wireless channels. The routing performance are influenced by transmitted traffic due to the following reasons:

1. Increasing the network traffic results in reduction of network lifetime: both sender and receiver sides of communication need to consume energy to transmit network packets. Hence, increasing the network traffic results in increasing network energy consumption and consequently reducing network lifetime.
2. Increasing network traffic has the potential to reduce the routing throughput: it increases network congestion and message failure.

Figure 4-8: End-to-end delay (ETE) of client/server routing protocols.

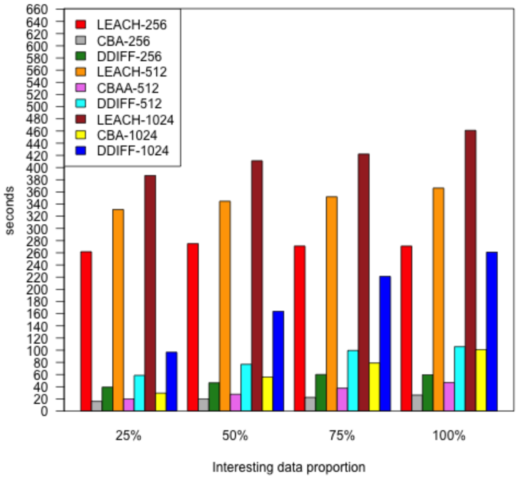


(a) small area (200×200)m^2

(b) medium area (400×400)m^2

Key points of the figures:

1. CBA outperforms the benchmark protocols in terms of ETE.
2. LEACH increases ETE in dense and/or large networks as compared to DDiFF.
3. DDiFF increases ETE in small and dense networks as compared to LEACH.



(c) large area (800×800)m^2

3. Increasing network traffic increases ETE: data packets need to be queued until the wireless channel become available (idle listening) if the network traffic increases. This means that increasing the network traffic increases waiting time for the sensor nodes to access the wireless channels for communication. The packets would be queued for a longer period until the wireless channel become available. Moreover, increasing the network traffic increases data packet failures and consequently increased data packet delivery time. The probability of message failure is increased due to message collisions when the network traffic rises. Hence, nodes need to re-transmit data packets until they are correctly delivered to the destination. It would increase the data collection ETE especially when the network is dense.

Evaluation: As Figure 4-9 shows CBA reduces the total transmitted traffic as compared to LEACH. Reducing clustering overhead and utilising the parallel guided-collision to establish the tree backbone are two key techniques that have the potential to reduce the network traffic in CBA.

1. Reduces clustering overhead: CBA dynamically groups the desirable source nodes into DC clusters according to the sink queries. It results in reduction of network traffic for two reasons: (1) reducing the cluster count in CBA: the number of clusters formed is reduced in CBA as compared to LEACH when data density is low in the network. In CBA, the clusters are formed if they are required to collect and aggregate interesting data samples for the sink. On the other hand, LEACH partitions the entire network into AC clusters without considering the content of available data at the nodes. All nodes need to collect and forward the required routing information to form the clusters. For this reason, a fewer number of clusters are formed in CBA as compared to LEACH especially when the number of desirable source nodes (data density) is low. (2) Periodical clustering and CH selection in LEACH: the CMs frequently need to transmit control packets periodically to select the new CHs in LEACH. It results in a significant increase in the number of transmitted control packets in LEACH especially when the network deployed is dense and large. Figure A-3 compares the total number of transmitted control packets in CBA and the benchmark data aggregation routing protocols.
2. Parallel guided-collision technique: utilising parallel guided-collision during the aggregation tree construction has the potential to reduce the number of control packets. The nodes avoid forwarding useless and/or redundant

messages from the sensor nodes to establish the BP paths to the sink. On the other hand, the route requests are heuristically forwarded by the nodes in LEACH to find/establish the required paths to route the data packets from the event regions to the sink. This results in increasing the transmitted network traffic especially when the area size increases.

A fewer amount of traffic is transmitted in DDiFF compared to CBA and LEACH, especially when the number of interesting source nodes and/or node count increases. DDiFF does not use a specific hierarchical infrastructure to aggregate and/or route the data packets. Hence, the sensor nodes do not need to forward control packets to establish/maintain such the hierarchical infrastructure to aggregate and forward data packets. On the other hand, CBA and LEACH need to forward a greater number of control packets to construct data aggregation infrastructures. This results in increased network traffic (and overhearing) especially when the network is dense.

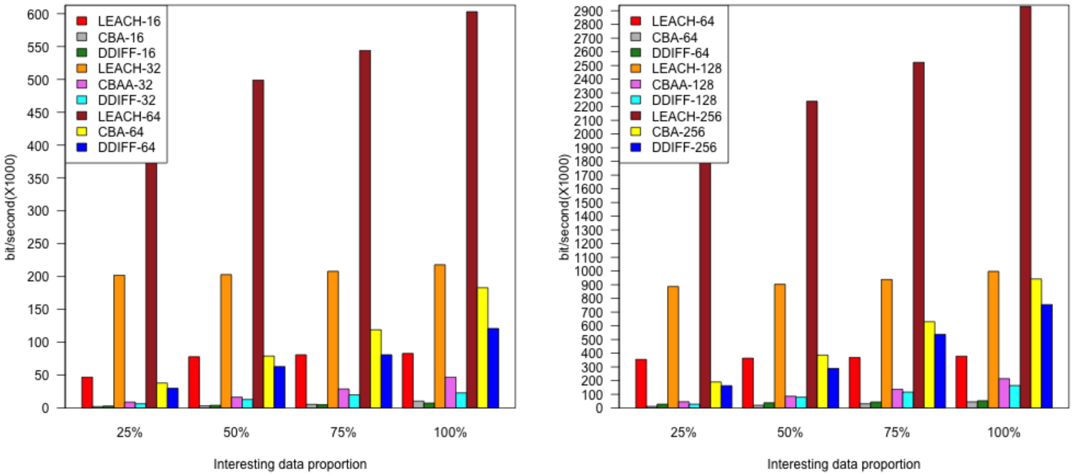
4.6 Discussion and Conclusion

CBA partitions the network into a set of DC clusters and then establishes a tree backbone to forward and aggregate the results of each cluster to the sink. The performance of CBA does not depend on the distribution model of event sources (ER or RS) and node heterogeneity in the network. This means that CBA is able to collect, aggregate and report desirable data (according to the sink interests) from different sensor nodes which measure variant types of ambient data. CBA outperforms LEACH and DDiFF in terms of data aggregation accuracy and delay especially when area and node count increases. CBA consumes less energy than LEACH, but more than DDiFF. Forming DC clusters using the Hamming distance, establishing the spanning tree backbone, utilising parallel guided collision and hybrid routing are the techniques that enhance the performance of CBA in comparison to the benchmark protocols.

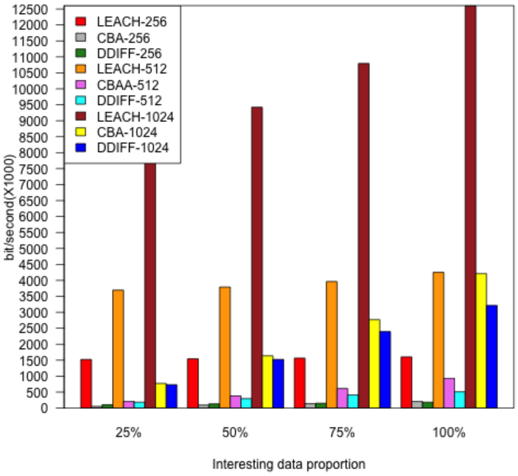
The performance of data aggregation routing depends on three metrics: energy, accuracy and delay. The metrics are influenced by communication type (single/multi-hop), routing scheme (proactive/reactive), node and data density. According to the results just presented, the metrics are correlated. The potential correlations between energy, accuracy and delay metrics are investigated and discussed in Chapter 5.

From the results, it is observed that CBA sharply enhances data aggregation accuracy as compared to the benchmark protocols. This is for two reasons: reducing the probability of message collision and forwarding the data results through reliable paths.

Figure 4-9: Total transmitted traffic of client/server routing protocols.



(a) small area (200×200)m^2 (b) medium area (400×400)m^2



(c) large area (800×800)m^2

Key points of the figures:

1. CBA transmits less network traffic as compared to LEACH.
2. Network traffic in LEACH is significantly increased when the network is large and dense.
3. DDiFF transmit less network traffic as compared to CBA when data density in the network increases.

1. Reducing the probability of message collisions: CBA partitions the network into a set of DC cluster for data aggregation. Clustering localises the data communications into intra-cluster communications which result in reduction of message collision/failure. In other words, message collision is reduced in clustered network due to reducing the number of nodes which try to access the wireless channels to forward the data packets. On the other hand, the probability of message collisions are increased in a flat network in which the source nodes individually try to access the wireless channels to transmit their data samples. For this reason, the accuracy in flat routing protocols such as DDiFF sharply drops when the node count increases and the network is large and dense.

2. Forwarding the data packets through reliable paths: CBA forms a tree backbone to forward the aggregated result of each cluster to the sink. The tree branches are formed as the result of collided route requests which are forwarded from the CHs to the sink. Data packets are aggregated and forwarded over the tree until they reach the sink. On the other hand, the chance of data packet loss or failure is increased when packets are heuristically routed from the event regions to the sink. Without such infrastructure, the routing protocols (such as LEACH) need to establish paths dynamically to forward data packets to the sink. The data packets fail to be delivered due to node failure, lack of required energy to transmit the data packets, environmental noise and wireless communication intervention especially when the network is large and dense.

The end-to-end delay (ETE) is reduced in CBA as compared to LEACH and DDiFF. Decreasing the path hop count, reducing routing and access time delays are the key factors reducing ETE in CBA.

1. Decreasing the path hop count: CBA establishes a spanning tree using minimum hop count paths from the event regions to the sink. Each CH forwards the aggregated result of its cluster to the sink through shortest paths in terms of hop count. Hence, the communication delay caused by sending/receiving data packets at intermediate nodes on the paths reduces.

2. Reducing routing delay: data packets are proactively forwarded over the tree backbone from the CHs to the sink. The nodes which reside in the tree (TMs) do not need to collect the routing information to forward the data packets as they already have a reserved path to the sink. TM nodes aggregate the received data packets and forward the result to the next level of the tree using unicast. This results in a reduction of delay in comparison to the reactive routing protocols such as LEACH.

3. Reducing wireless channel access time: CBA reduces wireless channel access time by decreasing the number of nodes which try to access the wireless channels. This arises from supporting early data aggregation and DC clustering in the proposed data aggregation routing protocol. CBA allows the data packets to be aggregated on BP paths as soon as possible (using APC value). The data packets are forwarded from the CHs to the closest node (minimum hop count path) which reside in the tree infrastructure and has the potential to aggregate the data packets. Owing to this, data packets are hierarchically aggregated over the tree (at MPs) and then forwarded until the sink is reached. It reduces the number of data transmissions and consequently the number of required relay nodes as compared to the routing protocols which heuristically forward the data packets and/or aggregate the data packets late. Moreover, forming DC clusters reduces the cluster count when the number of desirable source nodes is low in the network. DC clusters are formed on-demand according to the sink interests. A source node joins a DC cluster if it matches the data consumer interests. Consequently, fewer clusters are formed in DC in comparison to AC clustering that covers whole the network without considering the content of available data at the nodes. As a result, the requests to access the wireless channels are reduced as the number of CHs is reduced. It results in increasing the availability of wireless channels and consequently reduction in access time.

CBA enhances energy efficiency in comparison to LEACH. Similar to CBA, LEACH is a routing protocol which establishes a hierarchical infrastructure (clusters) to support data aggregation in WSN. It forms the clusters in a AC manner in which the CHs collect intra-cluster data packets to aggregate and report to the sink. CBA reduces the energy consumption as compared to LEACH due to the following reasons:

1. Reducing the clustering cost: CBA utilises the Hamming distance to partition the network into the clusters. The clusters are formed in a DC manner using local communications according to the data consumer interests. The sensor nodes join a cluster if they have data which match the sink queries. On the other hand, LEACH forms the clusters in AC manner in which all the network nodes need to participate in the clustering procedure. Moreover, the sensor nodes periodically need to broadcast control packets to collect the routing information which are required to select new CHs and form the clusters. Hence, energy consumption of LEACH is increased, especially when the node count increases. Figure A-3 compares the transmitted control packets in CBA and the benchmark protocols during the data aggregation procedure.

2. Reducing the data collection cost: CBA forms a spanning tree to report the collected and aggregated intra-cluster data packets to the sink. The tree allows the data packets to be aggregated hierarchically and forwarded to the sink. In other words, the tree reduces the number of data packets which are forwarded from the event regions to the sink. Moreover, the tree infrastructure is established using an energy-efficient technique which is called parallel guided collision. This technique has the potential to reduce energy consumption for establishing the tree infrastructure as it avoids to forward/broadcast useless/redundant control packets. Consequently, the cost of reporting data packets from the event regions to the sink is reduced as compared to LEACH which heuristically route the data packets from each cluster to the sink.

CBA consumes more energy than DDiFF because of the cost of establishing the hierarchical infrastructure for data aggregation. CBA forms the clusters to reduce the probability of message collision/loss and consequently improve data aggregation accuracy (Figure 4-6). As a result, a correlation is potentially exist between energy consumption and accuracy. This correlation will be investigated and discussed in Chapter 5.

In conclusion, CBA generally has a better performance compared to LEACH and DDiFF in either large or small network with variant node count and data density. It has the ability properly to collect, aggregate and forward the desirable data samples which are scattered across the network either in RS (random detection) or ER (100% detection) models. The performance of CBA is better as the area size, node count and/or data density get higher. The performance of CBA does not depend on the node heterogeneity in terms of data and communication ability (uni or bi-directional communications). However, CBA is not as efficient as DDiFF in terms of energy consumption according to the results. It seems that the capacity for CBA to collect, aggregate and report a greater number of data samples is potentially the reason for increased energy consumption compared to DDiFF. This issue is investigated in Chapter 5 by considering the potential correlations between energy, accuracy and delay.

4.7 Summary

A cluster-based routing algorithm, called CBA, is proposed to support data aggregation in WSNs. CBA aims to maximise energy efficiency and data aggregation accuracy, and minimise end-to-end delay. According to the results, a satisfactory performance of CBA is observed that satisfies its objectives as compared to LEACH and Directed Diffusion routing protocols in which client/server model of data aggregation is supported. Dynamic data-centric clustering gives CBA the ability to collect and aggregate desirable data samples regardless of their distribution model and/or type heterogeneity. This means that CBA has the potential to work over WSN in which sensor nodes are equipped with multiple sensing modules (e.g. TelosB mote) to measure a range of ambient events that may be distributed in the sensing field either in RS or ER model. CBA reduces energy consumption as it forms clusters using the lightweight Hamming distance technique. This avoids involving sensor nodes whose data is not interesting for the data consumer to collect. Moreover, utilising collision guided technique reduces the number of transmitted control packets and consequently decreases energy consumption during the routing infrastructure establishment phase. The end-to-end delay of data aggregation is reduced in CBA by avoiding random routing and establishing shortest paths (minimum hop count) from the data regions (clusters) to the sink. In addition, reducing routing delay by utilising a proactive routing approach over the tree backbone to route data packets helps in reducing ETE. The accuracy of data aggregation is enhanced in CBA as the ratio of data packet delivery is increased. Reducing the message collision/loss due to reducing the network traffic, forming data centric clusters and establishing a network backbone to route data packets, which has the effect of enhancing accuracy in CBA.

Chapter 5

The Correlations between Performance Parameters

This chapter focuses on the existing correlations between a selected set of data aggregation routing metrics namely energy consumption, accuracy and delay. The correlations between energy consumption, accuracy and delay can be examined in the form a triangle in which each edge focuses on a potential correlation between two metrics. First, energy-delay which focuses on the minimising energy consumption as much as possible with respect to delay. Second, energy-accuracy that considers maximising the data aggregation accuracy while minimising the consumed energy. Third studies the correlations between delay and accuracy to minimise the end-to-end delay while maximising the data aggregation accuracy. Energy-accuracy and energy-delay trade-offs have been previously studied in [Boulis et al., 2003], [Solis and Obraczka, 2006] and [Lindsey et al., 2002]. In addition, this research investigates the correlation between delay and accuracy. The performance of the proposed and benchmark routing protocols in both routing models namely mobile agent (Chapter 3) and client/server (Chapter 4) are evaluated and discussed according to the existing correlations in the triangle. Section 5.1 introduces the chosen data aggregation routing metrics. It highlights the key issues which may influence each of the parameters during data aggregation routing in WSN. Section 5.2 presents and explains the potential correlations between each pair of parameters. The correlation results for mobile agent and client/server models of routing are discussed in Section 5.3. Section 5.4 discusses and summarises the correlation results to highlight and compare the key issues of the trade-offs in both routing models.

5.1 Routing Performance Metrics

The results from Chapter 3 and 4 show a set of potential correlations between the performance parameters. From the various metrics on which we collected data in Chapters 3 and 4, we focus on energy, delay and accuracy here, because as vital as energy is for WSN, delay and accuracy are critical in terms of meeting consumer requirements for data aggregation requirements, such as maximisation of accuracy or minimisation of delay. Studying the correlations is intended to highlight the challenging issues that influence the performance of data aggregation routing in WSNs.

5.1.1 Energy Consumption

Power consumption is a factor in designing WSN protocols as the sensor nodes are highly energy limited. The sensor nodes are often powered by no more than a couple of AA batteries. Re-charging the sensor node batteries is entirely impractical if the network is deployed in a hostile and/or harsh environments. Hence, sensor nodes are unreachable either for changing or re-charging batteries. For this reason, an over-riding objective of WSN protocols is to find suitable solutions which save and/or balance energy consumption and consequently enhance network lifetime.

Sensor nodes consume energy for data processing, wireless communication and/or sensing data. Table 5.1 shows the estimated currents requirements to perform sensor tasks for the Mica2 mote [Landsiedel et al., 2005]. The network energy is consumed on both sides of the communication (sender and receiver) according to a wireless energy consumption model. The model consists of two parts reflecting transmission and reception as shown in equation 5.1 and 5.2 respectively [Wang et al., 2006]. A sender node needs to consume energy E_{TX_e} to run the transmitter circuit and E_{amp} to activate the transmitter amplifier, whereas a receiver consumes E_{Rx_e} power for running the receiver circuit. Energy consumption in wireless communication also depends on message length (K). Besides, transmission distance (d) have a high impact on energy consumption on the sender side that means more energy is typically required to transmit messages over greater distances. As a result, it can be concluded that energy consumption in WSN depends on four features [Basurra, 2012]: (1) the node type (Network Interface Controller features), (2) packet size, (3) communication distance and (4) transmitted network traffic (usage of the network bandwidth).Accordingly, and in light of the equations 5.1 and 5.2, energy consumption should be reduced if size/number of network packets and/or the length of communication paths reduces. For this reason, utilising efficient routing techniques to establish shortest paths between event regions and the sink, and supporting in-network data aggregation all have the potential to reduce

Table 5.1: Currents for Mica2

CPU	**Current**	**Radio**	**Current**
Active	7.6 mA	Receive (Rx)	16.4 mA
Idle	3.3 mA	Transmit (Tx)	17 mA
ADC Noise	1.0 mA	setup Tx/Rx	8.2 mA
Power down	116 μA	**Memory**	**Current**
Standby	237 μA	Read	4 mA
Sensor board	0.7 mA	Write	15 mA

energy consumption in WSNs.

$$E_{TX}(k,d) \approx (E_{TX_e} \times K + E_{amp} \times K \times d^2) \tag{5.1}$$

$$E_{RX}(K) \approx (E_{RX_e} \times K) \tag{5.2}$$

Generally speaking, there are four key issues which have the potential to waste network energy (at the MAC layer) in WSNs [Demirkol et al., 2006]:

1. **Overhearing:** receiving packets which are destined for other nodes is called overhearing. It has the potential to significantly increase the network energy consumption in WSNs especially when message broadcasting is used, because sensor nodes consume energy to receive messages which are not addressed to them.

2. **Packet collision:** sender nodes have to re-transmit messages in the case of packet collisions. As the corrupted/collided messages are usually discarded in WSN communications [Sohraby et al., 2007], the sender nodes need to re-transmit the failed messages until they are properly delivered to the receivers, resulting in increased energy consumption.

3. **Control/data packet overhead:** network energy consumption is highly dependant on the number and/or size of control/data packets. This means that network energy consumption increases as the number and/or size of transmissions increases.

4. **Idle listening:** energy consumption increases if sensor nodes frequently (or continuously) listen to the wireless channels to receive the packets that have yet to be sent [Hohlt et al., 2004].

Utilising efficient communication paradigms (in routing protocols) has the potential to reduce the impact of these issues on energy consumption. For example, forwarding network messages in unicast or multicast instead of broadcast reduces energy consumption as this reduces the number of nodes receiving network packets and hence overall network energy consumption. Moreover, utilising low-cost MAC protocols in WSN reduces communication overhead by reducing idle-listening, message collisions and/or the size/number of control packets [Ye et al., 2002], [Miller and Vaidya, 2005]. For example, B-MAC [BONNY, 2005], which is a MAC protocol specifically designed for WSN, typically reduces energy consumption by reducing idle-listening cost as compared to the S-MAC mac protocol [Ye et al., 2002].

5.1.2 End-to-end delay

Average end-to-end delay (ETE) is measured as the average time from the start of the data aggregation routing procedure until the (aggregated) data samples are delivered to the sink. Data aggregation routing would start at a specific time which is set by the data consumer (i.e MAs activation or sensor nodes report time). This finishes for each (aggregated) data packet or MA upon reaching the sink. Reducing ETE is an objective of WSN data aggregation routing protocols as it contributes to data collection freshness. Decreasing delay gives the system the ability to provide real-time data collection services for the data consumer. However, out-of-date and/or expired data samples are collected, aggregated and delivered to the sink if the delay is high.

Data aggregation ETE is measured differently depending on the data aggregation model. In the MA model, ETE is measured as the average journey time of all received MAs at the sink. The number of MAs active during data aggregation (being the degree of parallelism) and the delay caused by MA itinerary planning and migration are the most influential parameters for MA data aggregation end-to-end delay.

1. **Multiple MAs:** utilising multiple MAs for data aggregation reduces ETE by collecting and aggregating data samples in parallel. However, increasing the number of MAs may negatively affect network energy consumption and/or congestion. This is because the communication overhead and network traffic (to transmit the MAs) increases in line with the number of MAs.

2. **Itinerary planing delay:** MA itinerary planning delay has the potential to increase ETE. It stems from the delay in collecting routing information required to compute the MA itineraries. The information can be collected either in a proactive or reactive manner. Proactive MA itinerary planning reduces ETE (as compared to reactive) because the MA routing information is already collected at the visited nodes. However, the cost of proactive itinerary planning goes up if the network topology changes frequently because network resources are consumed to

update the proactive routing information. The cost is increased depending on the size of the regions in which the nodes proactively collect the routing information. This means that the cost of updating the proactive routing information is higher in larger proactive regions because a greater number of nodes need to update their routing tables.

3. **MA migration delay (communication delay):** it depends upon the effectiveness of the MA routing algorithm to discover/establish shortest paths for MAs migration. The migration delay depends on the number of nodes residing on the MA migration path. It can be reduced if the routing algorithm establishes a minimum hop count path for the MA to move. Indeed, increasing the hop count increases the communication delays to send/receive the MAs and consequently increases the migration delay.

Client/server ETE is measured as the average time from when the source nodes start forwarding data samples until the aggregated results are received by the sink. The delays caused by communication, in-network aggregation and data packet routing are the metrics which have a high impact on client/server ETE.

1. **Communication delay:** ETE is increased depending on the communication delays at the intermediate nodes which receive and forward the data packets between the event regions and the sink.

2. **In-network data aggregation delay:** it increases ETE. The intermediate aggregators wait for a while until the data samples are received form the neighbour nodes. For example, a parent node residing on a aggregation tree needs to wait until data samples are received from the child nodes for aggregation. Hence, ETE increases depending on the number of source nodes connected to the intermediate aggregators. In-network data aggregation delay increases when the network deployed is dense, as each intermediate aggregator needs to wait to receive data packets from a greater number of source nodes.

3. **Routing delay:** it depends on the efficiency of the routing algorithm to route the data packets from the source nodes to the sink. ETE decreases if the delay caused by routing information collection, path establishment and route length is reduced.

Communication delay is viewed as the main parameter affecting ETE in both data aggregation models. It depends on the nature of wireless communication and can be reduced by decreasing the network traffic and/or path hop count. There are six key factors which influence communication delay [Maroti et al., 2004]: (1) Send time: measured as the time to transfer messages to the node's MAC layer on the transmitter side. It depends on node CPU loads and operating system. (2) Access time: is the required time to access the wireless channels. It depends on network traffic. (3) Transmission time: time for transmitting a message at the transmitter side. It depends on message length. (4) Propagation time: measured from when a message leaves the transmitter wireless channel until is received by the receiver wireless channel. It depends on the distance between the nodes. (5) Reception time: the time of message receipt by the MAC layer at the receiver side. Similar to transmission time, it depends on the message length. (6) Receive time: the queueing time in which the messages are enqueued at the receiver side until they are processed. It depends on the queue length, CPU loads and the node operating system.

5.1.3 Accuracy

The number of data samples which are properly delivered to the sink defines the *accuracy* [Boulis et al., 2003]. Maximising accuracy gives the data consumer the possibility to make better decisions on the collected data. This means that using a greater number of data samples in data aggregation enhances data aggregation robustness and provide a more precise summary for further analysis. For this reason, a key objective of WSN data aggregation routing protocols is to maximise accuracy.

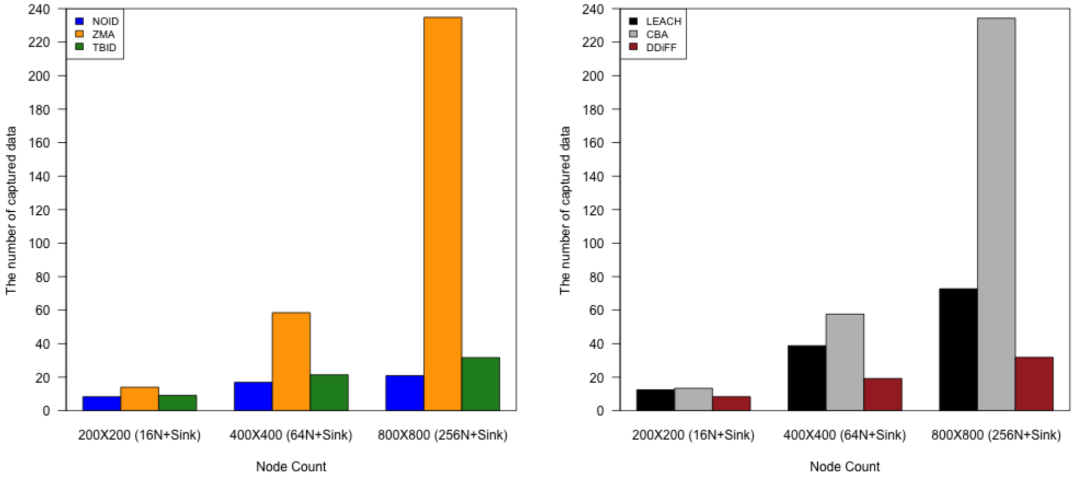
Accuracy is defined and measured depending on the model of data aggregation. In the mobile agent model, accuracy is defined as the number of data samples which are picked up by the MAs during their journeys. The effectiveness of the MA data aggregation protocols to find the data regions, collect the interesting data samples and return the results to the sink is the issue which has the most potential to influence the accuracy. This means that the accuracy is increased if the routing algorithms are able to discover the interesting data regions to forward the MAs for data aggregation. However, the MAs may miss a set of source nodes to visit if the algorithm is not able properly to find the data regions. In addition, the accuracy of MA data aggregation fall if the MAs are lost due to the ineffectiveness of the routing algorithm that should guide them back to the sink. In the client/server model, data aggregation accuracy is measured as the number of data packets which are aggregated and delivered to the sink. Accuracy is reduced due to data packet collision/loss. It also depends on network traffic and/or effectiveness of the routing algorithm. This means that fewer data packets are delivered to the sink or intermediate aggregators for aggregation if they collide or are lost. Increasing network traffic enhances the probability of wireless message failures due to the increasing the number of nodes which (simultaneously) try to access wireless channels. Traffic is reduced by grouping the source nodes to reduce the number of simultaneous wireless transmitters. In other words, grouping the nodes would result in reducing the number of sensor nodes which try to access wireless channels because only the group representatives (i.e cluster-heads) would stay in duty of the grouped nodes communications. MAC protocols also have the potential to reduce the number of message collisions/losses because they balance transmitter node access to the wireless channels. In addition, the performance of routing algorithm influences the accuracy. This means that data aggregation accuracy is increased if the routing protocol is able to discover/form data regions and report the data packets through reliable paths to the sink. According to the experimental results in Chapter 3 and 4, it is observed that our proposed algorithms (ZMA and CBA) enhance accuracy especially when the experiment size increases.

5.1.4 Accuracy versus Experiment Size

The accuracy in Chapter 3 and 4 is measured according to the experiments in which node count, data density and area size varies. All those experiments counted the sink as one of the nodes which resides in the middle of area. Hence, the number of randomly-placed sensor nodes varies in 15/31/63 ($200 \times 200m^2$), 63/127/255 ($400 \times 400m^2$) and 255/511/1023 ($800 \times 800m^2$). However, this is not quite in proportion.

To investigate the relationship between accuracy and experiment size further, we changed the setup slightly to have a constant ratio of randomly-placed sensor nodes to the experiments. This means that, we need to add one extra randomly-placed node to each scenario to have 16/32/64 ($200 \times 200m^2$), 64/128/256 ($400 \times 400m^2$) and 256/512/1024 ($800 \times 800m^2$) sensor nodes plus one sink. We measure then the accuracy of the proposed protocols and benchmarks in a selected set (sparsest network) of the experiments when the experiment size varies in proportion. In other words, the accuracy is measured for three (sparse)networks including 16 ($200 \times 200m^2$), 64 ($400 \times 400m^2$) and 256 ($800 \times 800m^2$) randomly-placed nodes plus a sink which is positioned in the middle. The sink has no sensor data to participate in data aggregation, whereas 100% of randomly-placed sensor nodes have desirable data to collect. As Figure 5-1 shows, the accuracy of ZMA and CBA increases (very similar to the results

Figure 5-1: Accuracy vs. Experiment Size.



(a) Mobile Agent Protocols

(b) Client/Server Protocols

of Chapter 3 and 4) when the experiment size increases quite in proportion.

5.2 The Correlations between Data Aggregation Routing Parameters

This section explains the correlations between data aggregation routing parameters namely energy consumption, delay and accuracy. According to these correlations, a trade-off triangle is formed which has the potential to evaluate the performance of the data aggregation routing protocols. Each edge of the triangle focuses on the existing/potential correlation between each pair of the routing parameters. The triangle edges are discussed in the following sections.

5.2.1 Energy-Delay Correlation

Consumed energy is correlated with end-to-end delay in data aggregation applications. This means that energy saving results in increasing ETE. Increasing the number of intermediate nodes on a path reduces energy consumption as the communication distance between the nodes reduces. According to equation 5.1, the energy consumption at transmitter side is correlated to (approximately) the square of the distance. On the other hand, increasing the path hop count results in increased communication delays and consequently increased ETE. For example, according to Figure 5-2, let us assume that node X wants to send a message to node D. Two paths are offered to forward the

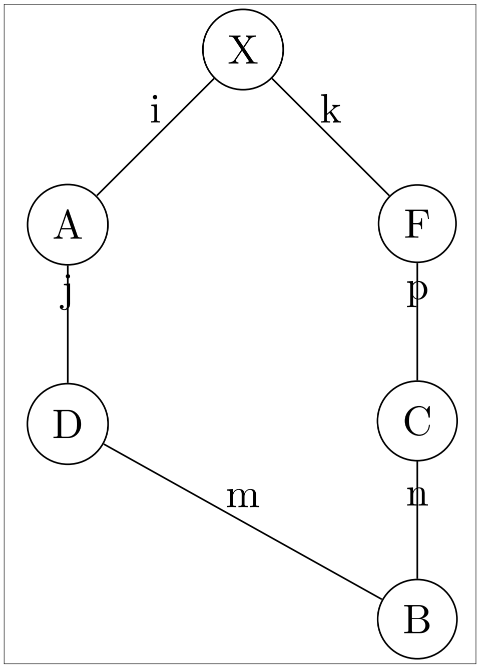


Figure 5-2: An example of Energy-Delay Trade-off

packet: first via node A and second through nodes F, C and B. It is assumed that the label of each edge shows the path length (distance) and (i+j) is greater than (k+m+n+p). If node X selects the former path, the total energy consumption is greater than the F-C-B route because the path length (euclidean distance) is longer. On the other hand, ETE is increased due to the communication delays at the intermediate hops on the F-C-B path. This means that reducing energy consumption by establishing multi-hop paths to reduce the communication distance results in increasing ETE. Owing to this, a routing protocols would be efficiently evaluated if the correlation between energy consumption and ETE is considered. According to [Lindsey et al., 2002], the value of energy×delay is recommended for evaluating the efficiency of routing protocols.

5.2.2 Energy-Accuracy Correlation

The energy-accuracy correlation arises from the energy consumed for collecting and reporting the data samples [Boulis et al., 2003]. Energy saving is a vital issue in WSN for the maximisation of network lifetime, whereas increasing accuracy is extremely desirable for the data consumer to enhance the robustness of collected data. Energy consumption increases if a greater number of data samples are delivered to the sink using either client/server or MA. This is because of the increased number of data samples which are forwarded to the sink, either as data packets or MAs, and hence increase energy consumption. On the other hand, the routing protocols consume network energy to route data samples which may never be delivered to the sink (failed data packets or lost MAs). Owing to this, the performance of a routing protocol would not be efficiently tested and/or evaluated if only one of energy consumption or accuracy is considered. The energy-accuracy correlation seeks to maximise accuracy with respect to energy consumption. Consequently, the energy required to deliver one data sample is measured by evaluating the routing protocol's performance based on the correlation [Boulis et al., 2003] and shows the capacity of a data aggregation routing protocol to save energy in the collection and aggregation of data samples.

5.2.3 Delay-Accuracy Correlation

The performance of data aggregation routing protocols would not be properly evaluated (in terms of latency) if only ETE is considered. ETE measures the average received time for data packets and/or MAs at the sink. Hence, a routing protocol has a lower ETE if only a few number of data samples (using data packets or MAs) are delivered to the sink. In other words, achieving a lower ETE does not necessarily mean a better performance for a data aggregation routing protocol if a number of data samples fail to be delivered (collided data packets or lost MAs) to the sink. On the other hand, it seems that increasing the data accuracy by collecting a greater number of data samples at the sink would increase ETE. In fact, increasing the number of data transmissions or the nodes which participate in data aggregation routing would increase network traffic and communication delays, that consequently results in higher ETE. For this reason, the performance of data aggregation routing protocols needs to be evaluated according to the correlation between ETE and accuracy, in which the protocol's ability to minimise ETE while maximising accuracy is examined. However, the correlation between ETE and accuracy seems to be complex. This means that ETE is not directly correlated to accuracy as it depends on a range of additional parameters such as MAC protocol, the degree of parallelism in data aggregation (i.e the number of active MAs), network traffic, path hop count and communication delays (according to the results of Chapter 3 and 4). ETE for one delivered data sample (delay-accuracy correlation) is considered in this chapter to investigate the potential delay-accuracy correlation and evaluate the performance of a data aggregation routing protocol according to the correlation.

5.3 Results

This section evaluates the performance of the routing protocols according to the existing correlations between energy consumption, accuracy and end-to-end delay. According to pervious studies [Boulis et al., 2003], [Solis and Obraczka, 2006] and [Lindsey et al., 2002], energy-delay and accuracy-energy correlations have been used to evaluate the performance of data aggregation routing algorithms. Owing to this, the proposed routing protocols in Chapter 3 (mobile agent) and 4 (client/server) are evaluated based the correlation metrics and compared then in each model of routing to the respective benchmark protocols. In addition, the average delay of each reported data sample is measured to investigate any potential correlation between delay and accuracy and evaluate the performance of data aggregation routing protocols.

5.3.1 Mobile Agent Data Aggregation Routing

This section studies the potential correlation between energy consumption, delay and accuracy in mobile agent data aggregation protocols. It evaluates the performance of ZMA (Zone-based Mobile Agent protocol), NOID [Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010] according to the correlation metrics.

5.3.1.1 Energy-Delay Correlation

The performance of ZMA is evaluated against NOID and TBID according to the result of energy consumption and delay. The value of energy×delay is influenced by a range of parameters such as overhearing, single/multi-hop communications, network density, transmitted traffic and routing scheme (reactive/proactive). The objective is to reduce the value of energy×delay as much as possible.

From Figure 5-3 it is observed that the result of energy and delay is increased when node count, data density and/or area size increases. We put forward the following reasons:

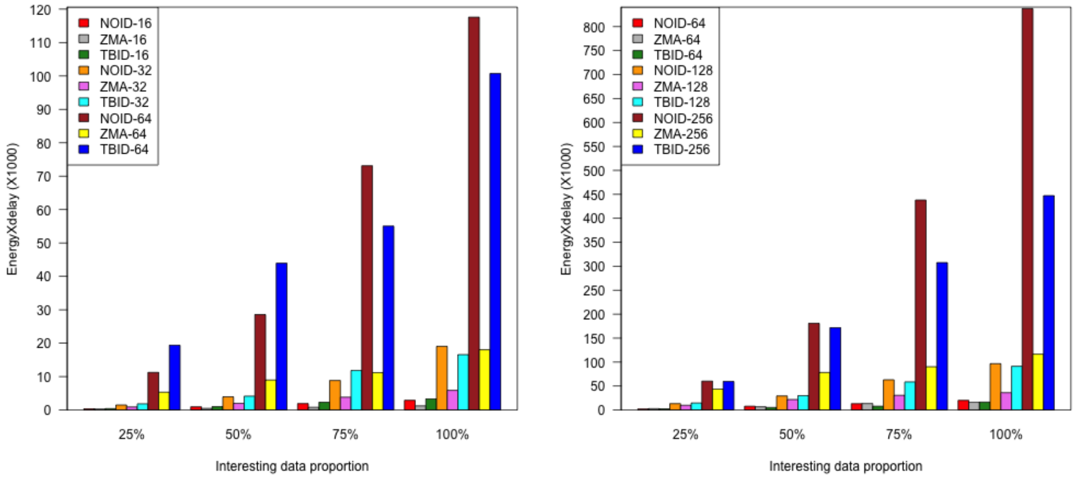
1. Increasing node count increases network density which results in increasing overhearing. The connectivity degree of the nodes is increased if a network becomes denser. Hence, a greater number of nodes receive/overhear the control packets which are broadcast to route the MAs. For this reason, network energy consumption is increased. In addition, increasing node count increases the itinerary planning delay. Each sensor node needs to collect a greater amount of routing information in its neighbourhood to route the MAs when the node count (or connectivity degree) increases. The impact of node count on ETE is higher in reactive routing protocols (i.e NOID) in comparison to proactive ones as the routing information needs to be reactively collected to route each MA. ETE is increased depending on the connectivity degree at each node which participates in MA routing.

2. Increasing data density increases the value of energy×delay. It depends upon increasing the number sensor nodes which try to join the data aggregation routing procedure as they have interesting data to report. In addition, a greater number of paths is offered to forward the MAs in the network when data density is enhanced. Hence, the network energy consumption and MA itinerary planning costs rise and would consequently result in increased energy×delay. The impact of the number of desirable source nodes on energy×delay would be higher in reactive routing (i.e NOID) as compared to proactive (i.e TBID). This is because of an increase in the itinerary planning delay in reactive routing when data density rises. The MAs would need to collect a greater amount of routing information at each node to establish best-fit routes to migrate, as the number of paths which are offered by the nodes is increased, when data density rises.

3. Increasing the area size increases energy×delay due to increasing path hop count between the event regions and the sink. Hence, ETE is increased (especially for reactive routing protocols) as the MAs would need to move through longer paths (in terms of hop count) to capture data samples from source nodes and return the collected data samples to the sink. Moreover, increasing path hop count increases the energy consumption as a greater number of nodes would participate in the MA routing, because nodes consume power to collect routing information, establish routing infrastructure (i.e TBID) and route the MAs. As a result, energy×delay is increased when area size increases.

As Figure 5-3 shows, ZMA has a better performance as compared to NOID and TBID especially when the area is not large. As mentioned earlier in Chapter 3, ETE and energy consumption is reduced in ZMA by avoiding blind/random migration, utilising hybrid routing and limiting the routing communication into the network zones. For this reason, ZMA has a better performance compared to the benchmark protocols in respect of energy consumption and delay. However, ETE is higher in ZMA when the network is deployed in large area (fig. 3-11c). ZMA is a hybrid routing algorithm in which proactive routing is used to route the MAs within the data regions, whereas reactive routing is used to forward the MAs between the data regions and the sink. For this reason, the MAs need to compute the return path on-the-fly at each TS (To-Sink) nodes to deliver the aggregated results to the sink. Owing to this, and due to the fact that the path hop count between the data regions and the sink increases in larger networks, ETE and consequently energy×delay is increased in ZMA as compared to proactive routing (i.e TBID). According to the figure, increasing the MA hop count increases ETE in ZMA as compared to NOID when a sparse network with low data

Figure 5-3: Energy×Delay in MA routing protocols.

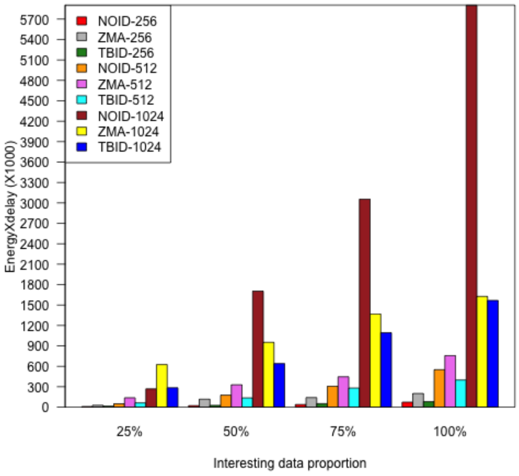


(a) small area (200×200)m^2

(b) medium area (400×400)m^2

Key points of the figures:

1. Node count, area size and/or data density influences energy×delay.
2. ZMA has a better performance in small and medium networks.
3. TBID reduces energy×delay if network deployed is dense and large.
4. energy×delay is significantly increased in NOID as the network becomes larger and/or denser.



(c) large area (800×800)m^2

density is deployed in a large area (fig. 3-11c). This stems from the ability of ZMA to find the source nodes which have desirable data samples to collect. According to Figure 3-10c, it is observed that the accuracy of ZMA is significantly increased as compared to NOID and TBID when the desirable source nodes are thinly scattered in a large area.

5.3.1.2 Energy-Accuracy Correlation

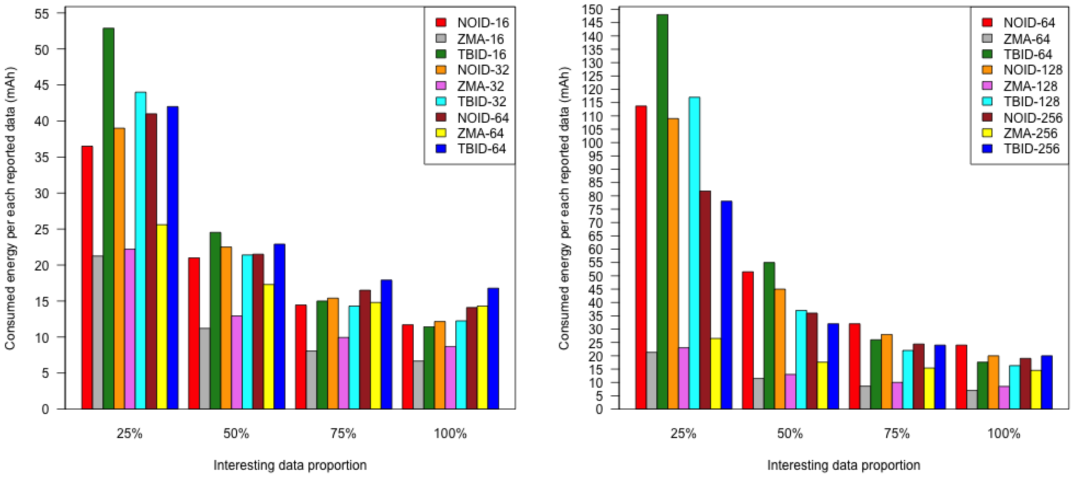
Energy consumption is influenced by the number of data samples which are captured and reported by the MAs. The consumed energy per each delivered data sample is measured to compare the energy-efficiency of the proposed MA data aggregation routing protocols (ZMA) to the benchmark protocols (NOID and TBID). It evaluates the routing protocols according to the amount of energy wasted in routing the MAs for data aggregation. Blind/random MA route search, routing scheme (proactive/reactive), overhearing and/or communication patterns (single/multi-hop) are the most influential parameters for the energy cost of reporting data in MA data aggregation routing.

From Figure 5-4, it is apparent that the consumed energy per each delivered data sample reduces when data density is increased in the network. It stems from the reduction in cost of reporting data samples in ER data regions in comparison to RS. Increasing data density (the number of desirable source nodes) increases the interconnectivity between the event regions that would result in forming data region which are mostly ER. Hence, the MA are able to find and capture data samples as the source nodes are interconnected and the MAs do not need to migrate randomly/blindly between the data regions to find the source nodes. In other words, the route search domain is reduced in ER event regions as the interesting routes are limited into the regions in which the interesting source nodes are interconnected. On the other hand, the cost of reporting data samples is increased when the MAs need to find interesting source nodes from the data regions which are thinly scattered across the network. The MAs need to migrate between the event regions to find and capture data samples which are interesting for the sink, but it would result in increasing the cost of reporting data sample to the sink. In NOID, forming the ER data regions would reduce the route search domain and/or blind/random migrations to find the desirable nodes in the network and it results a reduction of the cost of data reports when data density is increased. In TBID, a similar behaviour to NOID is observed according to the cost of ER/RS data aggregation. However, the energy cost of reporting data in TBID is increased when data density is reduced and data regions are mostly formed in RS. This is due to the cost of establishing the MA migration infrastructure and utilising proactive routing in TBID. The cost of reporting data is increased as the sensor nodes consume energy to establish the tree infrastructure and collect, maintain and update the routing information. Moreover, the sensor nodes in TBID would proactively collect the routing information that is never used by the MAs if the network deployed is sparse and they are isolated and/or not interconnected to the tree infrastructure.

ZMA has a better performance in reducing the cost of reporting data as compared to the benchmark protocols. It forms the data regions only by interconnecting the nodes which match the sink queries. In other words, ZMA allows the nodes which have interesting data samples to participate in collecting, maintaining and updating the routing information. Hence, ZMA avoids wasting network energy to collect routing information from nodes which are uninteresting and/or isolated. In addition, ZMA allocates a MA to each data region which then utilises a bottom-up data aggregation scheme to reduce blind/random migrations. In other words, the MAs avoid blind migration in ZMA and move throughout the network only if the paths guide them to desirable source nodes. Hence, ZMA pays the energy cost for data aggregation routing if a source node is visited. On the other hand, NOID reactively forwards the MAs through AC routes which are established at each node using the collected routing information from a limited area (single-hop neighbourhood). Owing to this, the MAs do not have an broad enough view to discover the data regions and capture the data samples. It increases the cost of data reporting in NOID as the MAs migrate blindly until source nodes are reached. TBID needs to establish a tree infrastructure in which the MAs move to capture and aggregate the data samples. Owing to this, each source node needs to establish a link to the tree to inform the MAs about its available data. It results in an increased cost of reporting data, especially when the network is sparse.

As Figure 5-4 shows, the performance of ZMA improves as the area increases. In large networks, communication between nodes are more likely to be multi-hop. Hence, the number of nodes which participate in the routing procedure is increased. It results in an increasing the cost of MA migration, routing infrastructure establishment and/or RS source node discovery. This means that the cost of reporting data is increased if a greater number of intermediate nodes consume the network energy for blind/random routing, collect useless (AC) routing information and/or establish the routing infrastructure from the event regions to the sink. For this reason, the energy cost of reporting data would be reduced if the number of node which participate in the MA routing is reduced. As ZMA forms the data regions and communication links in a DC according to the sink queries, the path hop count (or the number of node which participate in the MA routing) is reduced in comparison to AC routing especially in large networks.

Figure 5-4: Data reporting energy cost in MA routing protocols.

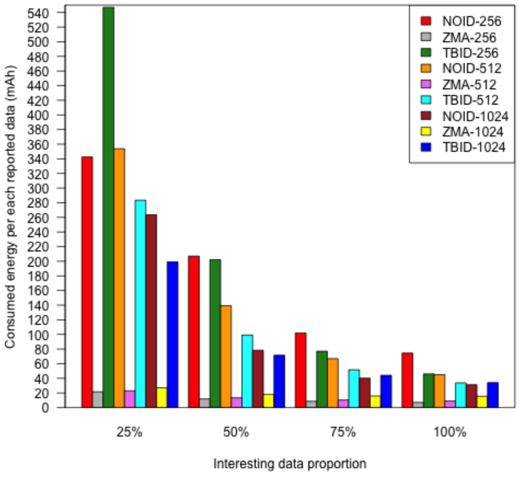


(a) small area (200×200)m^2

(b) medium area (400×400)m^2

Key points of the figures:

1. The energy cost of data reduces if data density is increased.
2. ZMA has a better performance as the area increases.
3. The energy cost in TBID is greater than NOID when the deployed network is small and sparse.
4. The energy cost in NOID is greater than TBID when the deployed network is large and dense.



(c) large area (800×800)m^2

5.3.1.3 Delay-Accuracy Correlation

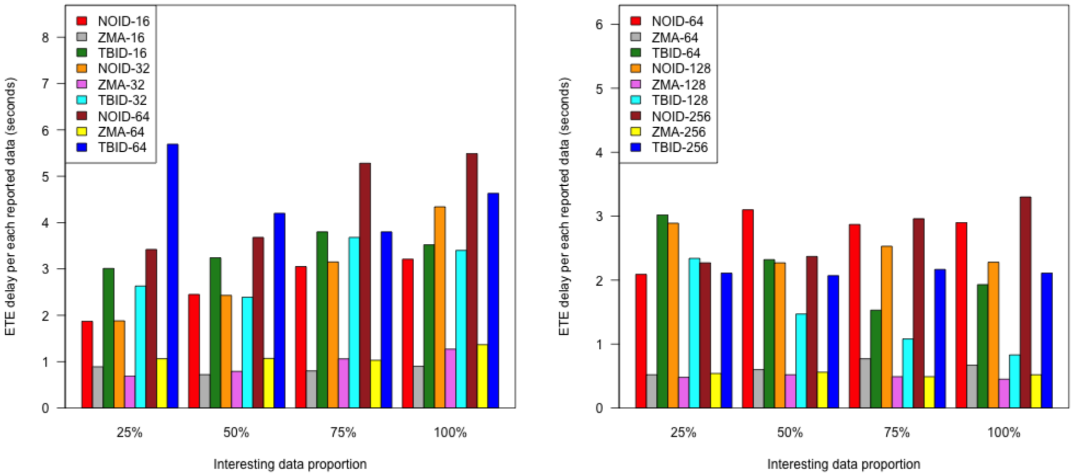
The correlation between delay and accuracy is investigated by measuring ETE for one delivered data sample in MA data aggregation routing. According to Figure 5-5, it is observed that ETE has a complex behaviour with respect to accuracy in the MA model of routing. It is because of that ETE is not directly correlated to the number of delivered data samples as it is influenced by a range of parameters such as the number of active MAs (the parallelism degree of data aggregation), MA hop count, the intelligence of itinerary planning, routing scheme (proactive/reactive) and/or single/multi-hop communications. In addition, ETE is influenced by a range of external issues like environmental noise and/or jamming attacks. For example, jamming attacks increase the wireless channel access time for the sensor nodes which want to form data regions or routing infrastructure, and/or route the MAs [Xu et al., 2005]. For this reason, variant behaviours of the delay-accuracy correlation are observed for the MA routing protocols if data density, node count and/or area size varies. Investigating about the impacts of the parameters on delay-accuracy correlation is addressed as a future work which will be discussed in Chapter 6.

To evaluate the performance of MA data aggregation routing protocols, Figure 5-5 shows that ZMA reduces ETE for each reporting data as compared to NOID and TBID. As discussed earlier in Chapter 3, ZMA limits the network traffic (especially control packets) into the data regions and communication zones to reduce the access time delay (fig. 3-13 and fig. A-2). It avoids blind/random migration and forwards the MAs through the minimum hop count paths from the data regions to the sink (fig. 3-12). Moreover, hybrid routing information collection reduces the itinerary planning delay as compared to reactive protocols (fig. 3-11). In addition, the number of MAs is sharply increased in ZMA when the area size increases (fig. A-1). In larger networks, a greater number of data regions is formed that results in increasing the number of MAs. Increasing the number of active MAs enhances the parallelism degree of data aggregation which results in reduction of end-to-end delay. For these reasons, the average delay of ZMA is reduced in comparison to the benchmark protocols.

5.3.2 Client/Server Data Aggregation Routing

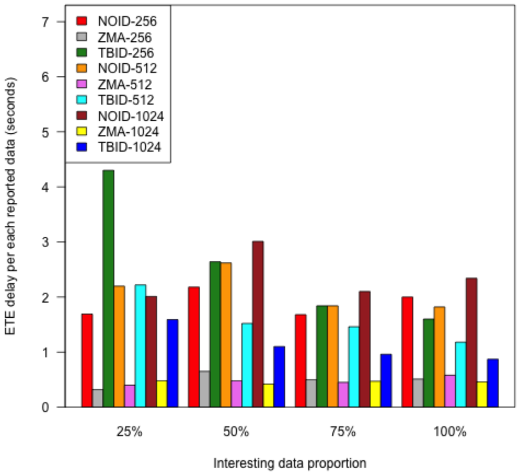
This section measures and studies the potential correlations between energy consumption, accuracy and end-to-end delay in client/server data aggregation routing similar to MA model. According to the results, the performance of Cluster-Based Aggregation protocol (CBA) is evaluated in comparison to the benchmark protocols namely LEACH [Heinzelman et al., 2000] and Directed diffusion (DDiFF) [Intanagonwiwat et al., 2000].

Figure 5-5: Delay-Accuracy correlation in MA routing protocols.



(a) small area (200×200)m^2

(b) medium area (400×400)m^2



(c) large area (800×800)m^2

Key points of the figures:

1. ZMA reduces the result of delay-accuracy as compared to the benchmark protocols.

5.3.2.1 Energy-Delay Correlation

Energy×delay shows the ability of routing protocols to reduce/balance energy consumption with respect to ETE. The performance of a routing protocol improves if energy consumption and data collection delay is reduced. Routing scheme (proactive/reactive), network traffic, node count and network size, overhearing and communication patterns (single and/or multi-hop) are the most influential parameters for energy consumption and/or ETE.

According to Figure 5-6, energy×delay is increased when node count, data density and/or area size increases. It is because of three reasons as below:

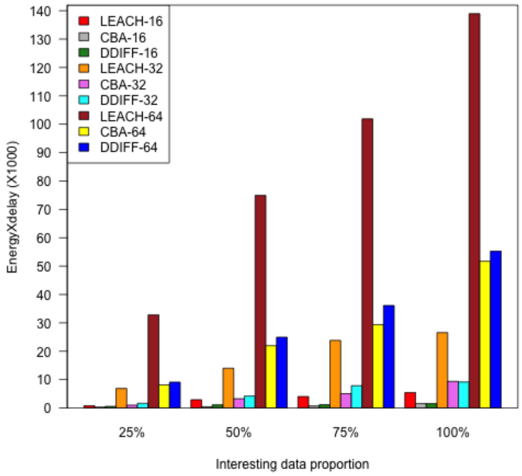
1. Energy×delay increases due to increasing the connectivity degree between the nodes when node count is increased. Increasing the degree of connectivity results in increasing overhearing as the nodes would be able to receive a greater number of transmitted messages from their neighbour nodes. It also increases the network energy consumption. In addition, the access time delay is increased as the availability of wireless channels is reduced when the node count rises, because of the increased number of requests to access the wireless channels. This means that ETE is increased as a greater number of nodes need to access the channels to forward data or control packets when the node count rises.

2. Data density influences energy×delay. Increasing data density results an increase in the number of nodes which participate in routing and/or data aggregation. It leads to an increase in network energy consumption and ETE.

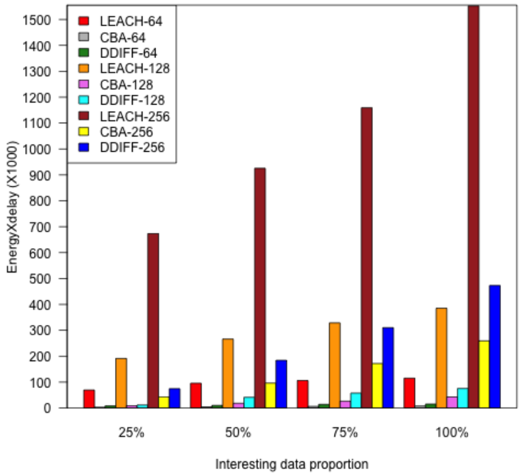
3. Increasing the network area increases energy×delay. Increasing the area results in increasing the path hop count between the event regions and the sink. It increases the number of nodes which participate in intermediate routing of data packets. Consequently, the energy consumption and ETE (communication delays) rise. The impact of area is higher on energy×delay when a reactive routing such as LEACH is used. In a large network (for example 800×800 m^2), the communication links are mostly established as multi-hop between the source regions and the sink. Hence, a greater number of sensor nodes need to heuristically route the data packets which are forwarded from the event regions to the sink. As a result, energy consumption and delay in reactive routing is significantly increased when area increases.

As Figure 5-6 shows, CBA reduces energy×delay as compared to LEACH especially when the area increases. CBA is able to route data packets from the event regions to

Figure 5-6: Energy×Delay in Client/Server routing protocols.



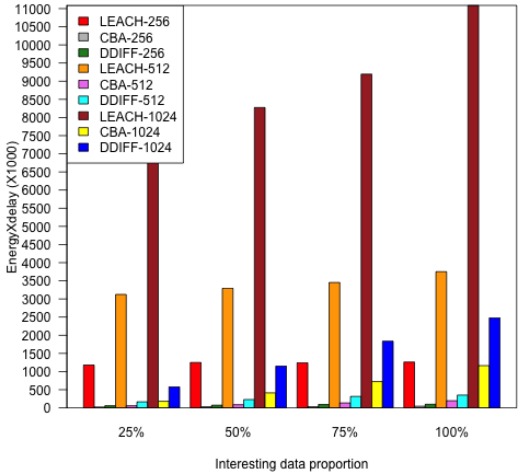
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

Key points of the figures:

1. Node count, area and/or data density influence energy×delay.
2. CBA performs better than LEACH and DDiFF especially when area size increases.
3. DDiFF reduces energy×delay in comparison to LEACH.



(c) large area (800×800)m^2

the sink via single/multi-hop links which minimise energy consumption and end-to-end delay. This stems from forming the DC clusters, establishing the minimum hop count paths (BPs) between the event regions and sink and utilising hybrid routing. The BPs (Backward Paths) are the minimum-hop-count routes which are established as a spanning tree from the event regions to the sink. They are in charge of routing and aggregating the data packets in a convergent manner from the CHs to the sink. Hybrid routing and minimising the hop count on the BPs results in a reduction of ETE as compared to LEACH in which data packets are reactively routed to the sink. In addition to reducing ETE (fig. 4-8), CBA reduces energy consumption in comparison to LEACH. CBA forms the clusters with the nodes which have desirable data samples to report. In other words, it avoids consuming energy to cluster nodes which have useless or uninteresting data samples based on the sink queries. On the other hand, LEACH clusters whole the network in a AC manner without considering the content of available data at the sensor nodes. It increases energy consumption and consequently energy×delay especially when data density is low. In addition, periodical AC clustering results in increasing the energy consumption in LEACH when the node count is increased. In fact, the number of sensor nodes which participate in AC clustering is increased in larger/denser networks, as LEACH needs to select periodically the CHs and/or form the clusters. Due to these reasons, energy×delay of LEACH is extremely increased when area increases.

CBA has a better performance in comparison to DDiFF in terms of reducing energy×delay when area increases. Firstly, this is a consequence of DDiFF increasing end-to-end delay in large networks. DDiFF forwards data packets through the paths which are individually established from each source node to the intermediate aggregators and/or sink. The access time and communication delays are increased due to increasing the number of nodes (residing on the variant paths) that simultaneously need to access the wireless channels to forward the data packets. On the other hand, CBA supports early data aggregation in which data packets are collected and aggregated from each DC cluster with minimum delay. The CHs forward the aggregated result of their intra-cluster data samples to the sink through the spanning tree which supports hierarchical data aggregation. It results in better availability of wireless channels as the number of transmitted data packets and/or relays nodes is reduced. Although DDiFF consumes less energy compared to CBA (according to Figure 4-5), increasing ETE in DDiFF leads to an increase in energy×delay.

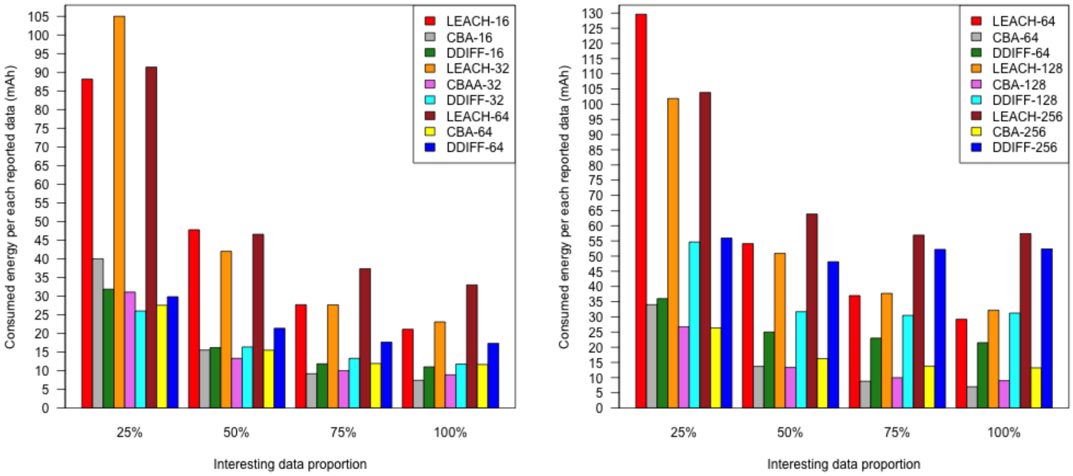
5.3.2.2 Energy-Accuracy Correlation

The number of reported data samples has the potential to influence energy consumption in data aggregation routing. Energy consumption of a routing protocol is increased if the routing protocol aims to collect, aggregate and report a greater number of data samples. For this reason, considering only the consumed energy is not an efficient way to evaluate the performance of a data aggregation routing protocols. The consumed energy per each delivered data would show the ability of routing protocols to avoid wasting network energy during data aggregation procedure. In other words, a routing protocol is more energy-efficient if it is able to minimise the average consumed energy per each delivered data. This section evaluates and compares the performance of CBA, DDiFF and LEACH according to the energy cost of reporting data. The routing scheme (address/data centric), routing proactivity/reactivity, the cost of establishing and/or maintaining the routing infrastructure, routing efficiency to establish reliable and connected paths and/or performing early/late data aggregation are the parameters that have the potential to influence the correlation between energy and accuracy in client/server routing protocols.

From Figure 5-7, it is observed, in line with intuition, that the consumed energy per each delivered data sample is generally reduced when data density increases in the network. This is due to the impact of the event source distribution model (ER or RS) on the consumed energy per reported data sample. The event regions in ER are largely formed where the number of source nodes is high, whereas RS event regions are established when the proportion of desirable sources is low. In other words, increasing the number of desirable source nodes results in increasing connectivity amongst them and consequently forming ER event regions. On the other hand, the event regions in RS are formed where the desirable source nodes are not interconnected as they are thinly scattered in the network. The cost of reporting data is reduced in the ER event regions as the data packets have a higher chance to be aggregated as early as possible. This results in reducing the number of transmitted data packets and required relay nodes to route the aggregated results. However, data packets need to be individually forwarded from variant parts of the network to the sink when the event regions are formed in RS. They are likely to traverse longer paths (in terms of hop count) until they get aggregated, especially when an AC routing approach such as LEACH is used. Accordingly, the consumed energy for each reported data is increased when the event regions are formed in RS and the interconnectivity between the desirable source nodes is low.

According to these results, CBA reduces the energy cost of reported data in comparison to DDiFF and LEACH, especially when the area increases. Utilising lightweight

Figure 5-7: Data reporting energy cost in Client/Server routing protocols.

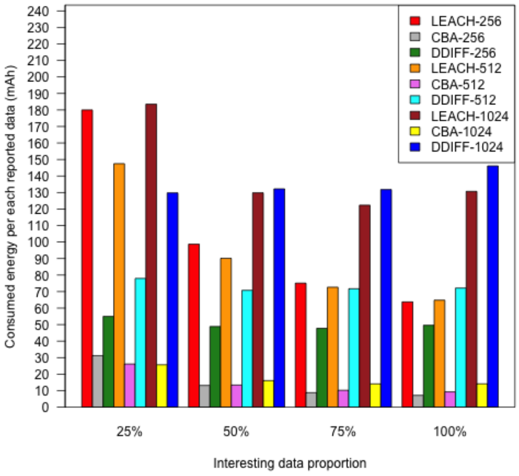


(a) small area (200×200)m^2

(b) medium area (400×400)m^2

Key points of the figures:

1. The energy cost of reporting data reduces if data density increases.
2. CBA performs better as the area increases.
3. The energy cost in LEACH is greater than DDiFF when network size is small and/or node density is low.
4. The energy cost in DDiFF is increased as compared to LEACH when the network deployed is large and dense.



(c) large area (800×800)m^2

and low cost techniques such as hamming distance and parallel guided collision instead of periodical clustering (LEACH) and/or query-based routing (DDiFF) to establish routing infrastructure would result in reducing the cost of reporting data. In addition, CBA saves network energy by avoiding inclusion of sensor nodes which do not match the sink interests. In other words, sensor nodes participate in clustering only if they have desirable data samples to report. It reduces establishment/maintenance cost of routing infrastructure (DC clusters and/or the spanning tree), especially when the node count increases. On the other hand, LEACH involves the whole the network in AC clustering that increases the cost of reporting data especially when node count and area increases. DDiFF does not form a hierarchical infrastructure for data aggregation routing and the data packets are forwarded through flat links to the intermediate aggregators and/or sink. Hence, a large number of individual paths need to be established from the source nodes, which are scattered in the network either in RS or ER, to the sink. This could lead to an increase in the number of nodes used for intermediate routing of data packets from the event regions to the sink. For this reason, the energy cost of reporting data is higher in DDiFF compared to CBA.

The cost of data reporting in DDiFF is lower compared to LEACH, as the proportion of desirable source nodes and/or area reduces. This is due to the cost of establishing and maintaing the hierarchical routing infrastructure in LEACH. LEACH partitions the whole network into AC clusters in which the CHs are periodically selected. This increases the number of control packets needed to form/maintain the clusters and select the CHs (fig. A-3). Increasing the transmitted control packets would increase the network energy consumption in LEACH. On the other hand, DDiFF does not establish a hierarchical infrastructure for data aggregation and route the data packets over a flat network from the event regions to the sink. Hence, data packets are delivered to the sink at a lower energy cost. On the other hand, the lack of a hierarchical infrastructure to localise the data transmissions and to reduce the number of transmitted data packets is the key reason for the increasing cost of reporting data in DDiFF as the area and data density increases. In DDiFF, each source node tries to forward its data packet to the sink through the reserved paths which are established during the route establishment phase. Hence, a greater number of data packets need to be forwarded by intermediate nodes to the sink as compared to LEACH, in which data packets get aggregated at the CHs. Owing to this, energy consumption is higher in DDiFF compared to LEACH when the deployed network is large and dense.

5.3.2.3 Delay-Accuracy Correlation

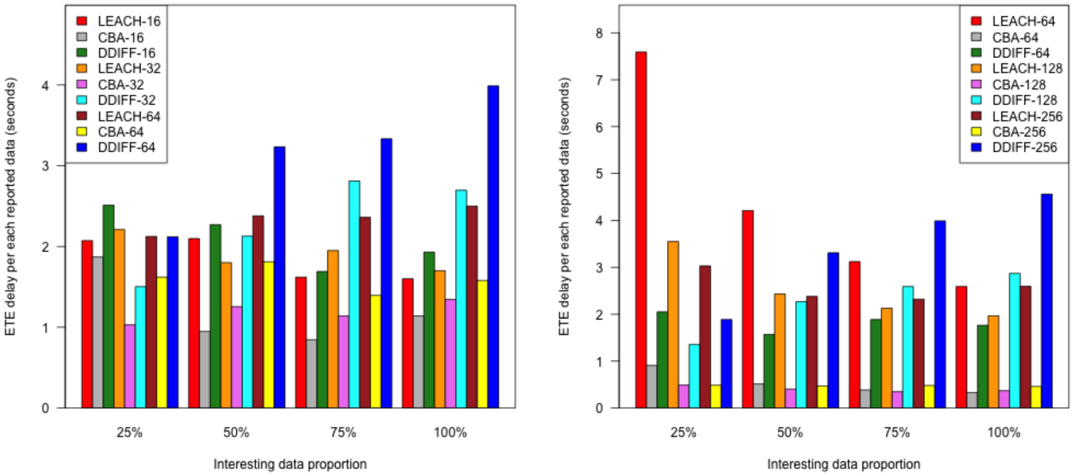
As it is apparent in Figure 5-8, delay-accuracy is improved in CBA compared to LEACH and DDiFF as node count and/or area increases. This is because of reduced end-to-end delay during the data aggregation procedure. Early data aggregation at DC CHs, utilising hybrid routing and uni-casting the data packets through the tree infrastructure from the event regions to the sink results in reduction of ETE. On the other hand, ETE is increased in LEACH due to reactive routing to report the data samples from the AC clusters to the sink. It is significantly increased as the area and path hop count between the event regions and the sink rises. ETE is increased in DDiFF as the data packets are individually forwarded through variant latency paths from each source node to the sink. This results in late data aggregation. Late data aggregation increases ETE, as the number of relay nodes and transmitted data packets increases if the node count and/or area increases.

According to Figure 5-8b and 5-8c, it is observed that the delay caused by reporting a data sample in LEACH is less than DDiFF when data density increases. This follows from the ability of LEACH to perform early data aggregation. A large number of data packets are aggregated at CHs in LEACH when the network has a large number of source nodes which have interesting data to report. On the other hand, each source node needs to forward individually its data samples to an intermediate aggregator or sink for aggregation in DDiFF. The data packets are delivered to the aggregators with variant delays as they are forwarded through variant hop count links from the event regions. Owing to this, DDiFF increases data reporting delay compared to LEACH when the network deployed is dense and large.

A data packet is delivered with a greater delay in LEACH compared to DDiFF when the interesting source nodes are thinly distributed in the network. This is due to forwarding the data packets from the CHs to the sink in a reactive manner. Data packets have a lower chance to be aggregated at CHs, as they are thinly distributed in the network and the clusters are formed in an AC manner. Hence, the probability of receiving multiple data packets at the AC CHs is reduced compared to a dense network with a large number of source nodes. Each CH needs to forward individually the data packets received from the CMs to the sink through the paths that are established reactively. This results in increasing LEACH's data reporting delay, compared to DDiFF in which each source node forwards its data through a proactively established (shortest) path to the sink.

Client/server routing protocols exhibit complex behaviours in respect of the delay-accuracy correlation if the network density, node count and/or area changes. It is because of that ETE is not correlated only to the number of delivered data samples.

Figure 5-8: Delay-Accuracy correlation in Client/Server routing protocols.

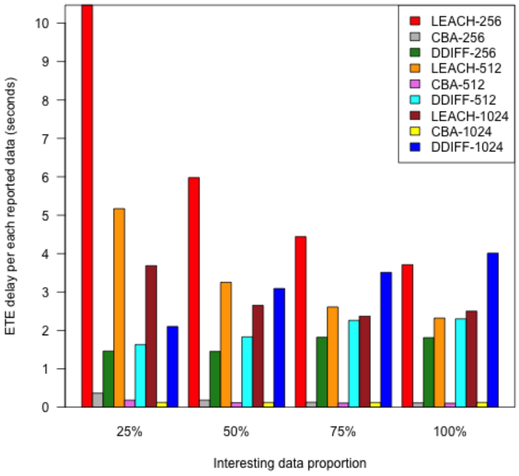


(a) small area $(200\times200)m^2$

(b) medium area $(400\times400)m^2$

Key points of the figures:

1. CBA improves the result of delay-accuracy as compared to the benchmark protocols.
2. DDiFF increases delay compared to LEACH in dense/large networks.
3. LEACH increases delay compared to DDiFF in sparse/small networks.



(c) large area $(800\times800)m^2$

ETE is influenced by a set of parameters such as communication delays (especially access time), path hop count, routing scheme (proactive/reactive) and/or single/multi-hop communications in client/server data aggregation routing. This issue is addressed as a future work in Chapter 6.

5.4 Conclusion and Summary

The performance of the proposed algorithms for both routing models (MA and client/server) have been investigated according to the existing/potential correlations between energy consumption, accuracy and end-to-end delay. The energy-accuracy and energy-delay correlations have been identified and analysed in previous works [Boulis et al., 2003], [Solis and Obraczka, 2006] and [Lindsey et al., 2002] to evaluate the performance of data aggregation routing protocols. In addition, we have studied here the correlation between delay and accuracy to investigate the impact of delivered data samples on ETE.

The correlations can be viewed as forming a triangle: first, energy-delay which is measured as the result of energy consumption and ETE per data aggregation round. The objective is to minimise energy consumption as much as possible with respect to data aggregation ETE. Second, energy-accuracy that is calculated as the energy cost for each reported data sample. The accuracy needs to be maximised while minimising consumed energy. Third, delay-accuracy is measured as the delay attributable to each reported data sample. It is desirable to minimise ETE while maximising data aggregation accuracy. This trade-off triangle has been used to evaluate the performance of data aggregation routing protocols in both mobile agent (ZMA, NOID and TBID) and client/server (CBA, LEACH and DDiFF) models.

From the results of mobile agent routing, it is observed that ZMA generally has a satisfactory performance according to the triangle correlations as compared to NOID and TBID. This is due to the capability of ZMA to minimise energy consumption and ETE, while maximising accuracy. The results are concluded as below:

1. **Energy-Delay:** increasing the network size and/or density increases the result of energy×delay. This is a consequence of increasing the number of nodes and/or established paths to which participate in MA routing. In proactive, the MAs utilise the collected routing information at each node to migrate, once the routing information is obtained. On the other hand, in reactive routing, they dynamically collect the routing information throughout the network until the sink is reached. Increasing network size and/or density affects reactive routing more than proactive, due to the increased amount of routing information which needs

to be collected dynamically. According to Figure 5-3, energy×delay is increased in NOID (reactive routing) more than TBID (proactive routing) as the network size and/or density increases. In other words, NOID has a better performance when the deployed network is sparse and small, whereas TBID reduces the result of energy-delay as the network becomes larger and denser. The performance of ZMA also is influenced by the routing scheme (reactive or proactive) as network size and/or density changes. ZMA utilises a hybrid scheme to route the MAs. The result of energy-delay is lower in ZMA compared to NOID when the network is dense and large. This is because it uses a proactive scheme (reducing ETE) to route the MAs within the data zones. On the other hand, energy×delay of ZMA is lower compared to TBID when the network deployed is small and sparse, because ZMA does not consume network energy to establish/maintain a proactive routing infrastructure to route the MAs from the event regions to the sink. By and large, the performance of ZMA is satisfactory compared to NOID and TBID in most the given scenarios. However, the performance of TBID is better than ZMA when a dense network is deployed in a large area (fig. 5-3c). This is because of reactive routing in ZMA to forward the MAs from the data zones to the sink. The path hop count between the data regions and sink is significantly increased in a large network, resulting in increased energy consumption and delay of (reactive) MA routing. Moreover, ZMA has a greater energy×delay result in comparison to NOID and TBID in a sparse network in a large area (fig. 5-3c). This is because ZMA finds and visits a greater number of desirable source nodes (as compared to NOID and TBID) to collect the data samples (fig. 3-10) using a bottom-up MA migration scheme.

2. **Energy-Accuracy:** the energy cost of reporting data reduces as the number of desirable source nodes increases. This is because of the decreasing energy cost of reporting data samples from ER data regions in comparison to RS. Increasing the degree of connectivity amongst the desirable source nodes in dense network enhances the ability of intermediate nodes to provide the routing information which is required for MAs to find the desirable nodes. Hence, ETE and energy consumption is reduced in ER compared to RS model in which source nodes are not interconnected. DC routing would be able to reduce the impact of the event source distribution models (ER/RS) on the energy cost of reporting data. This means that DC links would logically interconnect the random scattered (RS) event regions into ER zones. Hence, the MAs do not need to move dynamically/heuristically to discover the desirable source nodes. This should minimise the energy cost to find and capture the desirable data samples, compared to AC

routing protocols, especially when the network size and/or density rises. For this reason, the energy cost of reporting data in ZMA is lower in comparison to NOID and TBID (fig. 5-4).

3. **Delay-Accuracy:** the end-to-end delay is not directly correlated to the number of reported data in mobile agent routing. This is because of the impact of additional parameters such as the number of active MAs, network traffic, routing scheme (proactive/reactive) and/or communication type (single/multi-hop). However, the results show that ZMA reduces the delay caused by reporting a data sample, in comparison to the benchmark protocols. This arises from the capability of ZMA to reduce the MA itinerary planning and migration delays. ZMA does not forward the MAs blindly through the network to find the desirable source nodes and capture the data samples. The MAs are forwarded in ZMA through minimum hop DC paths which are formed according to the data consumer interests (fig. 3-12), which results in reduction of ETE. According to the results, it also can be observed that ETE is reduced in proactive routing (TBID) as compared to reactive (NOID), especially as the network size and/or density increases.

According to the results for client/server routing, CBA has a better performance based on the triangle metrics in comparison to LEACH and DDiFF. It shows the ability of CBA to achieve the objective of this research which focuses on minimising energy consumption and ETE while maximising the accuracy. The obtained results of client/server routing are concluded as follows:

1. **Energy-Delay:** energy×delay improves as node count and/or area size increases. Similar to MA routing, this is because of the increased number of nodes which participate in routing the data packets from the event regions to the sink. The impact of network size/density on energy×delay is greater in reactive client/server routing protocols in comparison to proactive ones. This is due to increasing the amount of routing information that is required to be collected during reactive routing. According to Figure 5-6, energy×delay rises in LEACH compared to DDiFF, especially when the area increases. LEACH forwards the aggregated results from each CH to the sink in a reactive manner. This results in increasing energy×delay when the network size increases. In addition, the cost of establishing/maintaing the AC clusters in LEACH affects energy×delay. LEACH partitions the whole network into AC clusters in which CHs are periodically re-selected. For this reason, the intermediate nodes need to transmit a large number of control packets (fig. A-3) to form the clusters, select the CHs and route the data

packets in each data aggregation round. The number/cost of the control packets is increased depending on the network node count and/or density. Hence, energy×delay is increased in LEACH as the network becomes denser and larger. On the other hand, CBA reduces energy×delay in comparison to the benchmark protocols, notably when network size and/or node density increases. This stems from two issues: first, CBA forms the clusters in a DC manner by involving the sensor nodes which have interesting data to report. This results in reducing the number of transmitted (sent/received) control packets during clustering and/or routing. Second, it utilises a hybrid routing approach to forward the aggregated results from event regions to the sink. This reduces the routing delay compared to reactive routing.

2. **Energy-Accuracy:** increasing data density in the network leads to a reduction of data reporting energy cost. This is a consequence of increasing the interconnectivity between the source nodes which have interesting data to report. The energy cost of reporting data is reduced as the data packets get aggregated earlier in a dense network, whereas in a sparse network, desirable source nodes may be isolated. The interconnectivity amongst the source nodes is increased if DC routing is used to establish the data forwarding paths. DC links are established according to the sink queries to connect the source nodes which are scattered in the network. Hence, data packets are aggregated earlier and at lower cost in comparison to AC routing. In consequence, the energy cost of reporting data is lower in CBA compared to LEACH and DDiFF. In addition, the results (fig. 5-7) show that hierarchical routing protocols (i.e LEACH) reduce data reporting energy cost in large and dense network, whereas flat routing protocols (i.e DDiFF) have a better performance in terms of energy-accuracy in small and sparse networks. LEACH has a higher energy cost of reporting data in small and/or sparse due to the cost of establishing/maintaining the hierarchical infrastructure routing. Conversely, DDiFF has a higher data reporting energy cost in dense and/or large networks due to the lack of a hierarchical infrastructure to localise data transmissions and perform early data aggregation.

3. **Delay-Accuracy:** delay is not directly correlated to accuracy as it is influenced by a range of additional parameters such as communication delays (especially access time), path hop count, routing scheme (proactive/reactive) and/or single/multi-hop communications. However, two key points can be observed from the results of delay-accuracy correlation. First, CBA reduces delay caused by reporting a data sample due to supporting early data aggregation at data-centric

CHs, establishing minimum hop count paths and utilising proactive routing to forward aggregated from the event regions to the sink. Second, establishing an hierarchical infrastructure for data aggregation routing (i.e LEACH) in large and/or dense network results in reduced delay in data sample reporting. This improves the chance of data packets to be aggregated sooner compared to flat networks.

In respect of these criteria, the ZMA and CBA protocols offer benefits over the established benchmark protocols for mobile agent and client/server routing schemes, respectively. They satisfactorily achieve the key objective of this research which is minimising energy consumption and end-to-end delay (ETE) while maximising the accuracy. Moreover, the impact of event source distribution model (ER/RS) and data heterogeneity on the performance of ZMA and CBA is reduced, compared to the benchmark protocols according to the results. The results of this chapter are concluded and summarised (for both models of routing) as below:

1. Increasing the network size and/or density increases energy×delay in data aggregation routing.
2. The result of energy×delay is increased in reactive routing in comparison to proactive as network size increases, whereas it is increased in proactive routing in comparison to reactive as network size reduces.
3. Increasing network (data and/or node) density reduces energy cost of reporting data.
4. DC routing reduces data reporting energy cost more than AC routing especially as network size increases.
5. Hierarchical routing reduces the data reporting energy cost in large/dense networks, whereas flat routing decreases it in small/sparse networks.
6. End-to-end delay is not directly correlated to the number of delivered data samples as it is influenced by a range of additional parameters, including path hop count, network traffic and routing scheme (proactive/reactive).

Chapter 6

Conclusions and Further Work

The widespread use of wireless sensor devices, and their advancements in terms of size, deployment cost, the ability to measure environmental events and user friendly interface have given rise to many applications of wireless sensor networks. Wireless sensor networks are usually characterised as self-organising networks which can be deployed without requiring any specific infrastructure or centralised control in harsh, hostile and/or inaccessible area. A WSN is deployed by means of a number of tiny, small and (relatively) cheap sensor nodes that are typically highly resource-constrained in terms of power, communication and computation. The key objective of WSN deployment is to collect ambient data with minimum cost. The measured data needs to be delivered to the sink for the data consumer via single or multi-hop paths which are established either in a proactive or reactive manner. The WSN technology has the potential to be used by rescue teams in the case of natural disasters such as earthquake. The sensor nodes can be scattered in the field (using a helicopter for example) for the purpose of finding/detecting victims. They can be deployed dynamically as a network in an ad-hoc fashion without requiring any existing infrastructure. Using the network, ambient data which is interesting for the rescue team (such as vital signs) can be measured and transmitted then from the environment to the data consumer access point (the sink). The trapped and/or injured victims can then be found and rescued, if their locations and vital status are properly reported to the sink.

WSNs need to utilise routing protocols to forward data samples from the event regions to the sink. The protocols are responsible for establishing the routes either in a proactive or reactive fashion similar to ad-hoc networks. However, as we said in Chapter 2 ad-hoc routing protocols are not effective for utilisation in WSNs due to sensor node resource constraints and WSN characteristics, specifically: (1) The lack of a global addressing scheme in WSN, (2) Sensor node resource constraints (mainly energy and bandwidth), (3) Convergent routing patterns in WSN to forward data from the data regions to the sink, (4) WSN routing is application dependent. As a result, a number of routing protocols have been developed with the characteristics of WSNs in mind, in which the focus is on how to forward collected data samples from the event regions to the sink via minimum cost links (in terms of energy consumption and time).

Data aggregation applications utilise routing protocols to collect, aggregate and deliver data samples in WSNs. The objective of data aggregation is to minimise the number and/or size of reporting data packets to save network resources – mainly energy. In fact, it aims to collect data samples from the source nodes and then combine them using the aggregation functions to express the information in a summary form for further analysis. There are two routing models for data aggregation applications: client/server and mobile agent (MA). Data samples are routed to either sink or intermediate aggregators via direct or indirect links in the former, whereas a single or multiple mobile agents traverse the network to capture and aggregate data in the latter.

There are three key parameters which have the potential to influence the performance of data aggregation applications in WSNs: energy consumption, end-to-end delay and accuracy. Energy consumption is a vital issue in WSN as it has a high impact on the network lifetime. It should be minimised to maximise the network lifetime. End-to-end delay is key to data freshness and should be minimised to provide real-time data for consumer analysis. Accuracy is measured as the number of data samples delivered to the sink. This should be maximised to enhance data robustness and to give the data consumer the ability to make better decisions on the basis of the collected data. As a result, minimising energy consumption and end-to-end delay while maximising accuracy is the collective objective of data aggregation routing in WSNs.

From reviewing the literature to identify where beneficial advances in the effectiveness of WSN routing could be made, this research was motivated to design and evaluate routing protocols for data aggregation in both models (client/server and mobile agent) with the primary objective of:

1. Collecting data samples from the event sources in RS and ER distribution models.
2. Collecting desirable data samples from (heterogeneous) networks in which the nodes may be able to collect variant data types.
3. Reducing energy consumption.
4. Minimising end-to-end delay of the data aggregation routing procedure.
5. Maximising the number of data samples delivered to the sink.

6.1 Summary and Conclusions

In Chapter 3, a mobile agent routing protocol (called ZMA) that uses a bottom-up approach for MA routing was presented and evaluated. Under this protocol, the network is divided into a set of contiguous zones to limit the communication overhead (reduce overhearing) and guide the mobile agents between the sink and event regions. At each zone, a set of data regions are dynamically formed in a data-centric manner according to the data consumer queries. Each data region is led by the most capable nodes in terms of having desirable data, maximum connectivity degree, minimum distance to the centre of region and the maximum level of residual energy. These nodes are called ZMACs and are identified by an autonomous zone construction process. The mobile agent migrations start from ZMACs to collect and aggregate data from the event regions to the sink in a bottom-up fashion. The mobile agent utilises a new data-centric itinerary planning algorithm in which the next nodes are selected according to a weighting function to maximise the number of captured data samples, avoid blind/random migrations and reduce the routing overhead (energy and delay). The MAs collect data samples from each data regions and return then to the sink via reactive paths to deliver the aggregated results.

The performance of ZMA is evaluated according to five metrics namely: total energy consumption, average end-to-end delay, MA hop count, total transmitted traffic and accuracy. Each experiment is run in 36 scenarios by varying node count (3 cases), area size (3 cases) and data density (4 cases) to test and evaluate the performance of ZMA. According to the results in Section 3.4, ZMA has a better performance compared to the benchmark protocols (NOID [Gavalas et al., 2010] and TBID [Konstantopoulos et al., 2010]) in most scenarios. The proposed algorithm improves on energy efficiency and data aggregation accuracy, especially as the area size and/or node density increases. In addition, it reduces the end-to-end delay as data density in the network increases. ZMA has an overall satisfactory performance and satisfies its objectives for the following reasons:

1. **Reduces routing overhearing:** ZMA localises the MA routing communications into restricted data regions which are dynamically formed in a DC manner. Indeed, ZMA allows the sensor nodes to collect the required routing information locally (in multi-cast or unicast) to forward the MAs at each data region. It reduces the communication overhearing as the routing communications in ZMA are restricted to the data regions in which the MAs are forwarded for data aggregation. On the other hand in NOID, the lack of a communication region results in broadcasting routing messages to collect routing information and discover/establish the

MA migration paths. It increases the transmitted (sent/received) network traffic and consequently the network energy consumption, especially when the network is dense.

2. **Intelligent MA routing:** ZMA avoids blind/random MA migrations and establishes only the paths which guide the MAs to the desirable source nodes. This means that the MAs in ZMA move throughout the network if a desirable source node is discovered to be visited instead of random walks. This results in a reduction of MA hop count, energy consumption and end-to-end delay. On the other hand, routing the MAs heuristically through address-centric paths (i.e NOID) increases path hop-count and consequently the energy consumption and delay as the MAs are forwarded blindly without considering available data content at the destination nodes.

3. **Bottom-up MA migration:** the MA journeys start to collect and aggregate data samples from the ZMACs residing in the centre of event regions. ZMACs have the maximum connectivity degree with the desirable source nodes at each data region. The intra-region routes are established in a data-centric manner in which each link is allocated by a weight metric according to DC connectivity degree and distance to the desirable source nodes. Through these routes, the MAs are forwarded in each data region to collect and aggregate data samples until the desirable source nodes are visited. On the other hand, the benchmark protocols start the MAs migration from the single-hop neighbours of the sink (top-down scheme) that are aware of the event regions. Hence, they miss the source nodes which fail to make a connection to the sink or routing infrastructure (i.e tree in TBID), especially when the event regions are formed according to a RS model. Owing to this, the number of captured data samples (accuracy) is reduced in the benchmark protocols as compared to ZMA when the network is large and sparse.

4. **Forms data regions:** ZMA forms a set of data regions by interconnecting the source nodes which have interesting data according to the sink queries. It limits the MA route search domain to the nodes which match the sink interests and are interconnected through DC links at each region. Hence, MA itinerary planning delay and consequently ETE is reduced, especially when the network is dense. Moreover, forming the data regions gives ZMA the ability to collect and aggregate data in both distribution models of event sources (RS and ER). The random scattered source nodes (RS) can be interconnected as (ER) data regions. Thus, capturing one source node from each data region results in visiting a number of source nodes in the network. On the other hand, the routing cost is significantly

increased in NOID (or TBID) when the source nodes are scattered in the network according to a RS model. Each MA needs to migrate from the sink to the RS event regions to capture the data and rerun the result. This results in increasing the routing path hop-count and ETE, especially when the network is large.

In Chapter 4, a client/server routing protocol (called CBA) that uses a hierarchical infrastructure for data aggregation is introduced and tested. CBA utilises the Hamming distance technique to partition the network in a decentralised manner into a set of data-centric clusters according to the sink interests. At each cluster a node which has the greatest level of residual energy is selected as cluster head to collect and aggregate intra-cluster data samples. The cluster-heads form then a spanning tree to the sink to forward and aggregate the data packets using a parallel collision-guided technique. This technique has the potential to decrease the routing overhead by reducing the number of control packet broadcast and avoiding forwarding unnecessary routing messages. Hop count and link cost (consumed energy) are reduced to establish low cost routes over the spanning tree.

CBA was tested and evaluated against the benchmark protocols in respect of five parameters: total energy consumption, total delivered data samples (accuracy), average hop count, average end-to-end delay and total transmitted traffic. As described in Chapter 4, the CBA experiments are run over 36 scenarios based on node count, data density and area. According to Section 4.5, the results of CBA meet the objectives of this research in comparison to the benchmark protocols namely LEACH [Heinzelman et al., 2000] and Directed diffusion (DDiFF) [Intanagonwiwat et al., 2000]. This is because CBA:

1. **Reduces message collisions:** CBA localises the data transmissions into clusters which are formed in a DC manner. This results in reducing the number of source nodes which need to access wireless channels to forward the data packets from the event regions to the sink, whereas, forwarding data over flat infrastructure (i.e DDiFF) results in increased message collision and consequently reduced accuracy as the number of source nodes increases.

2. **Reduces the establishment/maintenance cost of the hierarchical routing infrastructure:** CBA utilises the parallel collision guided technique to form the spanning tree from the event regions (CHs) to the sink to forward data packets. This technique is expected to reduce the cost of establishing the routing infrastructure as it decreases the number of routing (control) messages by avoiding to forward useless and/or redundant messages during the tree establishment phase. Moreover, CBA utilises the Hamming distance to form the DC clusters, so

that the network is partitioned into a set of DC clusters in a decentralised manner according to the sink queries. This reduces the clustering cost as compared to a routing protocol like LEACH which periodically partitions the whole network in an AC manner.

3. **Minimises the path hop count from the event regions to the sink:** CBA forwards the aggregated results from each CH through a spanning tree which is formed by the minimum hop count paths to the sink. This leads to a reduction in end-to-end delay and energy consumption in reporting data.

4. **Supports early data aggregation:** data packets are aggregated as soon as possible in CBA as the paths are established in a DC manner. Data packets are forwarded from source nodes to the DC CHs for aggregation. Then, the aggregated intra-cluster results are forwarded in parallel via minimum hop count DC paths to the closest (in terms of hop count) intermediate aggregator nodes which reside on the spanning tree. The results are hierarchically aggregated until the sink is reached. This leads to a reduction in energy consumption and end-to-end delay as compared to flat and/or AC routing protocols in which data packets are forwarded through (longer) links which do not consider data content during the data aggregation procedure.

In Chapter 5, the performance of the proposed algorithms in both MA and client/server mode of routing is evaluated in the context of the correlations between energy consumption, accuracy and end-to-end delay: (1) Energy-delay: the correlation between energy and delay is measured as the result of energy×delay in which the objective is to minimise the energy consumption with respect to the end-to-end delay (2) Energy-accuracy: this is calculated as the average consumed energy for reporting one data sample in the network; the objective is to minimise the consumed energy while maximising the accuracy (3) Delay-accuracy: this is measured as the average delay caused by reporting a data sample; this should be minimised to maximise data freshness and support real-time applications.

According to the results (Section 5.3), both proposed algorithms (ZMA and CBA) have a satisfactory performance in comparison to the respective benchmark protocols. The conclusions about ZMA and CBA performance are summarised below:

1. **Energy-delay:** (1) ZMA: has lower energy×delay in comparison to the benchmark protocols (NOID and TBID) for area up to medium size ($400\times400\ m^2$). This is due to forming DC data regions for MA migration and data aggregation, avoiding blind/random MA migration and reducing the transmitted network traffic. However, the result of energy×delay rises for ZMA when the network is large

(800×800 m^2). This is due to the ability of ZMA to find and capture a greater number of data samples during data aggregation routing as compared to the benchmark protocols. This means that the network energy consumption is higher in ZMA because of improved accuracy in comparison to NOID and TBID. In addition, ETE is higher in ZMA as MA itinerary planning delay goes up. Increasing the area results in increasing the path hop count which are established reactively from the event regions to the sink to forward the MAs. Indeed, ETE is increased as the MAs reactively collect the routing information at each intermediate nodes until the sink is reached. (2) CBA: has lower energy×delay as compared to the selected client/server routing protocols namely LEACH and DDiFF. Supporting early data aggregation, reducing the cost of establishing the routing hierarchical infrastructure and decreasing the hop count of data forwarding paths are the reasons for minimising energy×delay result in CBA.

2. **Energy-accuracy:** (1) ZMA: the energy cost of reporting data is lower in ZMA compared to the selected benchmark protocols. Data aware and intelligent MA migration, and the utilisation of a bottom-up scheme for the MAs to collect and deliver data samples are the key reasons for reduced data reporting energy cost in ZMA. In other words, the energy cost of reporting data is lower because ZMA does not waste network energy with blind/random migration and only forwards the MAs if there is a non-visited source node to visit. (2) CBA: reduces data reporting energy cost in comparison to the selected protocols (LEACH and DDiFF) especially as network size and/or density rises. This is because of the ability of CBA to reduce the probability of data packet collision/ loss by establishing the data aggregation tree from the event regions to the sink. In other words, CBA does not waste network energy establishing unreliable paths in which the data packets are lost or failed to be delivered. On the other hand, increasing the network traffic and/or utilising heuristic routing increases the probability of message failure in the benchmark protocols. A higher level of message failure increases energy consumption as the benchmark protocols consume network energy to report data packets which are never received by the sink. Hence, the cost of reporting data rises in the benchmark protocols, especially when the network is large and dense. However, DDiFF has lower energy consumption in comparison to CBA when the network is small and sparse. This is because it does not need to establish a hierarchical infrastructure to route packets. Hence, data delivery to the sink has a lower cost.

3. **Delay-accuracy:** both CBA and ZMA reduce the delay from reporting a data

samples in comparison to the respective benchmark protocols. Support for early aggregation (using DC links), decreased path hop count, reduced network traffic and hybrid routing are the key reasons for reduced delay in the proposed algorithms.

In conclusion, the performance of a data aggregation routing protocol is influenced by several factors: routing scheme (proactively or reactivity), event region distribution model (RS or ER), network size/density, data heterogeneity, routing infrastructure (flat and/or hierarchical) and routing type (data or address centric). The proposed algorithms (ZMA and CBA) resolve some of the problematic issues in existing data aggregation routing by eliminating/reducing the impact of these parameters on the routing performance. In other words, they have the ability to work in either large/small and/or sparse/dense network in which variant data types are measured by the sensor nodes from ER/RS event sources. Both ZMA and CBA significantly improve accuracy as compared to the respected benchmark protocols. Improving accuracy allows them to be feasible for data sensitive applications in which maximising the number of data samples at the sink is vital. In addition, they are able to support time-sensitive data aggregation applications due to the ability to reduce end-to-end delay. However, their energy efficiency needs to be improved, but that remains a topic for future work.

6.2 Contributions

The key contributions of this research to the current literature are:

1. Utilising bottom-up MA migration improves the performance of MA data aggregation routing as compared to top-down scheme. The MAs start their journeys from the centre of event regions to capture and aggregate data samples. This results in reducing the route search domain to discover the source nodes which have interesting data as compared to a top-down scheme in which the MAs move from the sink to the event regions. The DC links between the source nodes in each data region guide the MAs to the desirable source nodes. This reduces the routing overhead in collecting the MA itinerary planning information in comparison to top-down which needs to collect/search routing information of the event regions from the sink until the desirable source nodes are reached. For this reason, ZMA has a better performance in comparison to NOID and TBID which use a top-down scheme to move the MAs.

2. Forming data regions on-demand according to the sink interests improves the performance of MA data aggregation routing in the event regions which may be

scattered in a ER or RS model. The overhead of finding RS source nodes is significantly increased in MA routing especially when the network is dense and large. Utilising DC routing to interconnect the distributed source nodes and forming the data regions reduces the cost finding and capturing the source nodes. Indeed, the MAs capture the desirable source nodes through minimum hop count paths by which ETE and energy cost is also reduced. On the other hand, the MAs need to randomly/blindly migrate to/from different network parts until desirable source nodes are visited if no data region is formed in the network. It severely impacts energy consumption and delay, especially when only a few source nodes are interesting in a large and/or dense network (RS model).

3. Allocating the weight values to the paths according to the connectivity degree, residual energy level and distance to the data regions would enhance accuracy in MA data aggregation routing. Using the weight values, the MAs are forwarded through reliable paths (in terms of having sufficient energy) to the nodes which have the maximum connectivity and minimum distance to the desirable source nodes (data regions). Hence, the MAs do not need to migrate randomly/blindly but they are only forwarded if the destination node has desirable data to collect and/or knows minimum cost routes to the desirable source nodes. For this reason, the accuracy of ZMA is significantly improved compared to NOID and TBID.

4. Reporting data packets through a routing infrastructure (spanning tree) from data regions to the sink maximise accuracy and/or minimise delay in comparison to heuristic multi-hop routing in client/server data aggregation. Data packets collide or be lost when they are heuristically forwarded from the source nodes to the sink. Moreover, the end-to-end delay of reactive routing would be substantially increased especially when the network is dense and large. On the other hand, data packets can be transmitted through reliable and minimum hop count paths when a routing infrastructure is established from the event regions to the sink. The node residing on the routing infrastructure does not need to collect the routing information reactively and/or broadcast the data messages. This reduces network traffic and routing delay which consequently reduces the probability of message collision and end-to-end delay. CBA establishes a spanning tree from the event regions and the sink to forward the data packets.

5. The parallel collision-guided technique has the potential to reduce the establishment cost of routing infrastructure (tree). Although establishing a spanning tree from the event regions to the sink offers increased accuracy and reduced delay in client/server data aggregation routing, it is expensive in terms of en-

ergy consumption. Parallel collision guidance is a suitable technique to form the spanning tree that minimises the establishment cost. This technique avoids forwarding useless and/or redundant control packets which are forwarded during the tree establishment phase. Hence, the establishment cost of the tree is reduced as it is built using a fewer control packets than the benchmark algorithms (see Figure A-3). Consequently, the energy consumption does not rise significantly even if the network is dense and large.

6. The Hamming distance technique has the potential to form DC clusters in a decentralised manner with low energy cost. This is because it reduces the communication overhead during the clustering phase by reducing the size/number of transmitted messages. The sensor nodes only need to transmit a set of key features of the original measured data (i.e data type) that are required to form DC clusters. Moreover, they are able to eliminate redundant and/or useless data/messages. As the Hamming distance technique highlights the similarity of data, the source nodes are able to avoid transmitting same or similar data values which respectively have same or close Hamming distance. This results in a reduction of transmitted network traffic to form DC clusters. Consequently, CBA reduces the clustering cost in comparison to LEACH which periodically forms the clusters in AC manner.

7. Early data aggregation reduces energy consumption and end-to-end delay in client/server data aggregation routing. Aggregating data packets as soon as possible (in terms of traversed hop count) results in a reduction in the amount of transmitted traffic. Hence, a fewer number of relay nodes are required to relay the aggregated results from the event regions to the sink. On the other hand, the data packets are forwarded from source nodes to the aggregator nodes (i.e sink) when early data aggregation is not supported, which results in increased energy cost and end-to-end delay in client/server routing. For example, DDiFF establishes a set of separate paths from each source node to the sink or intermediate aggregators (route negotiator) to forward the data packets. Hence, the data packets need to be forwarded until they are received by the intermediate aggregators and sink to be aggregated. This increases the cost and delay of data aggregation routing especially when the network is dense and large. However, CBA supports early aggregation by combining the data packets at the CHs and/or MP nodes which reside hierarchically in the tree.

6.3 Future Work

This section outlines future work that has the potential to enhance the performance of the proposed routing algorithms, namely ZMA and CBA. Following on from the discussions in Chapter 5, these plans focus on four key issues, specifically: improving energy efficiency, enhancing data aggregation effectiveness, increasing the ability of dealing with network topology changes and further investigation of delay-accuracy correlation. These issues are addressed in the following subsections.

6.3.1 Improving Energy Efficiency

As mentioned in the summary and conclusion sections, the energy efficiency of both ZMA and CBA needs to be enhanced. Energy consumption can be reduced in the proposed algorithms using two techniques: utilising geographical information and wireless communication power adjustment.

Providing location information for the sensor nodes has the potential to reduce the energy consumption in CBA as the sensor nodes do not need to perform the cost allocation phase explained in Section 4.3.2. This would significantly reduce the transmitted traffic and consequently network resource consumption. The cost allocation phase is performed to find the distance of the source nodes to the sink. The distance to the sink allows the source nodes to avoid message broadcasting to establish the DC clusters, form the routing infrastructure (spanning tree) and/or forward the data packets. In ZMA, providing location information to the sensor nodes can reduce energy consumption for the following reasons:

1. The cost of ZMAC selection is reduced. Using the location information, the sink would be able to select directly the ZMAC nodes at the centre of each desirable region. For example, let us assume the sink query is to collect temperature data from zone 4 in a network. In the original ZMA, the source nodes which have temperature data to report (in any zone) try to select a set of nodes according to the ZMAC selection algorithm. The ZMACs which reside in zone 4 initiate the MAs to collect and then report the temperature data to the sink. On the other hand, the node which resides at the centre of zone 4 would be able to move the MAs to collect and report the desirable data, if the sensor nodes are equipped with a location information provider, such as GPS. Hence, the source nodes (residing in zone 4) do not need to communicate locally to select ZMACs at their neighbourhoods. This would reduce transmitted traffic and consequently energy consumption.

2. Using the location information, an optimal number of ZMACs can be selected to cover desirable data regions in the network. ZMACs are the nodes which stay on duty at each data region to manage the sink communications. This would result in a reduction of the communication overhead/traffic and network resource consumption (mainly energy) because the source nodes would be able to go to sleep until a wakeup message is received from the respected ZMACs. Consequently, selecting an optimal number of ZMACs to cover desirable areas according to the data consumer requirements would result in saving energy. In addition, the degree of parallelism arising from data aggregation routing can be managed according to the number of ZMACs in the network. The parallelism degree highly depends on the number of MAs which are generated/activated at ZMACs. Increasing the number of MAs would result in reduced end-to-end delay. However, it increases the energy consumption as the transmitted network traffic is increased. Owing to this, balancing energy consumption and delay (energy×delay) would depend on the number of MAs which move through the network to collect and report the data samples. Using the GPS-equipped sensor nodes, the sink would be able to select the exact required number of ZMACs which cover the desirable parts of the network and stay on duty to start the MA migrations. This would allow the data consumer to balance the degree of parallelism (the number of MAs/ZMACs) according to the data consumer requirements in terms of energy, delay and accuracy. This means that a greater number of ZMACs are selected if reduced ETE is required, whereas the number of ZMACs in minimised if energy consumption needs to be kept to a minimum.

Wireless communication power adjustment technique has the potential to reduce energy consumption in both models of data aggregation routing. This technique offers energy saving for WSN applications such as data aggregation in which sensor nodes frequently need to communicate. It saves network energy by adjusting the power of wireless signal at the source node according to its distance, which can be measured using RRSI [Xu et al., 2010], to the destination node. Sensor nodes consume energy according to an energy model (equation 5.1 that was explained in page 170 which highly depends on the distance. This means that a source node consumes a greater amount of energy if the communication distance is increased. Without wireless transmission power adjustment, a source node would use maximum power to communicate with any node which reside in its radio range. However, the destination nodes residing closer than the radio range border can be reached with less energy if the wireless transmission is accordingly adjusted. Hence, wireless transmission adjustment can be used to allow the source nodes to adjust their communication power according to their distance to destination node. As a results, the source nodes avoids transmitting at maximum power just to transmit control/data packets to nearby nodes. The results of the study in [Song et al., 2009] shows the network energy consumption is reduced when this technique is applied on WSNs.

6.3.2 Enhancing Data Aggregation Effectiveness

The performance of CBA can be improved if synchronisation is considered during data aggregation routing. Data packets are aggregated at the aggregator nodes and (hierarchically) forwarded until the results are received by the sink. However, reporting data samples without time synchronisation reduces the effectiveness of data aggregation. This means that the data aggregator nodes miss collecting and aggregating data packets which are received late. In this case, data packet are forwarded without aggregation to the sink. This would result in increasing the cost of data collection as the number of transmitted data packet is not as low as it might be. CBA partially resolves this issue using BHC (Backward path Hop Count) values which let the aggregator nodes (MPs of CHs) know the distance (in terms of hop count) to the CHs which have interesting data to report (or their RREQ messages are already received). In consequence, the aggregator nodes wait for a minimum required period, according to the link hop count to source CHs, to receive and aggregate the data packets. However, this technique needs to be optimised as the CHs are hierarchically interconnected to other child CHs whose data packets take a longer time to be received for aggregation. In other words, it is still a problematic issue if some of the data packets are received late (after the estimated/expected time which the aggregator nodes compute using BHC) due to the hierarchical links between the source CHs and/or MPs which stop forwarding collided RREQ messages in order to conserve energy. Synchronisation is commonly used to resolved this issue. However, it is expensive and consumes significant network resources especially when the network is large and dense. In addition, clock drift and skew errors may change sensor node clocks especially when the network is set up to work over a long period. Drift error is caused by environmental parameters like vibration, temperature, pressure, and battery voltage that have the potential to change the angular frequency of a node's clock oscillator, while skew error is the result of the different frequency of clock quartz crystals which may change the time [Ardakani et al., 2014]. Synchronising the aggregator nodes (i.e MPs) periodically is a potential solution to resolve this problem. It reduces the cost of (re-)synchronisation by limiting performance of synchronisation to the aggregator nodes instead of whole network. This means that the aggregator nodes residing on tree are hierarchically synchronised and then the aggregators at the higher level of hierarchy (with lower HC) would wait until the data packets from lower level ones (with higher HC) are received. This should result in efficient data packet aggregation and reduce the number of reported data packets – and hence the cost of data collection. However, the cost of synchronising the aggregator nodes would be still a problematic issue which leads to increases network resource consumption. It stems from a trade-off between the data aggregation effectiveness (by periodical synchronisation) and network resource consumption, and needs further investigation.

The Hamming distance technique has the potential to enhance the performance of CBA as it can be used to check data freshness and avoid reporting/collecting out-of-date data samples. Using this technique, the source nodes are able to find old data samples whose time values have a higher Hamming distance than recently sensed data samples. Hence, sensor nodes avoid transmitting old or expired data samples according to the data freshness requirements. It would result in a saving of network resources - primarily energy. For this reason, each data sample needs to be stamped with the time of sensing, whereas the data consumer should determine valid Hamming distance (time period) for data collection. The Hamming distance values can be frequently or periodically propagated by the sink according to QoS or data freshness requirements.

6.3.3 Increasing the Ability of Dealing with Network Topology Changes

The network topology can change in WSN as a result of node/link failures. A node fails in one of two ways: expected or unexpected. The node can be aware of failure (i.e running out of energy) and performs a set of failure recovery tasks in the former, whereas it fails suddenly (i.e hardware damage and/or node capture attack) in the latter. In addition, the network topology may change due to node mobility if the network is a MWSN. The possible techniques to deal with the network topology changes are discussed as below.

In the case of expected network topology change, CBA needs to perform different tasks depending on failing node/link roles in the network. The objective is to interconnect the maximum number of disconnected nodes caused by the node/link failures to the tree backbone. This means that CBA aims to keep the tree connected when any node residing on the infrastructure fails due to lack of/running out of energy. If the residual energy of node which resides on the data aggregation routing path falls below the threshold, the (failing) node would send a message ($Role_{Fail}$) to inform its interconnected neighbours, depending on its role. The neighbour nodes try to find/establish a link to the routing infrastructure using a message which is called Failed Route Request (FRR). A FRR is forwarded until is received by a node which reside on the backbone (a TM). At the end, the disconnected nodes are connected to the routing infrastructure when they receive a reply message from any node which is connected to the backbone.

Five examples (of which the first and second have already been implemented) are provided to show how CBA can deal with node failures depending on the role of the failing node in the network:

1. **A relay node failure:** relay node is a CM which is in charge of connecting a child CH to a parent CH (for example, node 22 in Figure 4-3). A failing relay node breaks its connections by sending a $Relay_{Fail}$ message to both parent and child CHs when its energy level falls below the threshold. Upon receiving the $Relay_{Fail}$ message, the parent CH removes the relay node's ID from its routing table. Besides, it forwards a FRR packet to establish a link to the disconnected child CHs. In other words, the FRR is forwarded to interconnect the parent CH to the child CHs via the intermediate nodes which have a greater HC (residing in a lower hierarchy level of the tree between the parent CH and disconnected child CHs) and/or a minimum BHC (the shortest path to the child CHs). The FRR is forwarded (through FPs) until is received by a node which maintains a RREQ from the disconnected child CH(s) in its routing table (i.e TM or a CM in child cluster). The node becomes the new relay node and forwards a version of RREP to the child CH(s) to interconnect them to the tree backbone. The RREP message with the minimum APC is used by the child CH if multiple RREPs through different links are received.

2. **A child CH failure:** a child CH is interconnected to any other cluster (a parent cluster) with a lower HC value (for example, node 29 in Figure 4-3). When the residual energy level of a child CH falls below the threshold, it sends a $Child_{Fail}$ message to its CMs to find a new CH. The $Child_{Fail}$ message includes the list of connected clusters in either lower or higher level of tree hierarchy (for example, cluster E and P if CH 29 in cluster S fails). The CMs reply back the failing CH by assigning their residual energy level and the list of requested clusters if there is a tie. The failing CH selects the node which has the most residual energy as the new CH. It also marks the CMs as relay nodes if they are connected to the clusters which have greater HC (residing in lower level of hierarchy). Then, it forwards a $Child_{BYE}$ message which includes the list of clusters with lower HC (residing in upper levels of hierarchy) and the list of relay nodes to the new CH. When the new CH receives $Child_{BYE}$, first, it updates its routing table according to the new relay nodes. Second, it tries to contact with the parent clusters that are listed in the $Child_{BYE}$ message using hello messages. If no response is received, the new CH forward a FRR (included by the list of CHs) to the TM nodes which have lower HC values if there is a tie. A TM node which knows the requested parent

CH forwards a RREP to the new CH. Hence, the new child CH is connected to the upper level clusters via the links, whereas it is interconnected to the lower level clusters through relay nodes.

3. **A parent CH failure:** a parent CH knows at least one cluster (a child cluster) with a higher HC (for example, node 17 in Figure 4-3). The failing parent CH tries to find a new CH amongst its CMs by sending a $Parent_{Fail}$ message similar the approach which is used in the case of a child CH failure. Then, the failing CH sends a $Parent_{BYE}$ message which is comprised of child CHs and TMs with lower (between the parent CH and sink) and higher (between the parent CH and child CHs) HC. The new CH, first, tries to find a CM which already has a connection to the TMs. If so, the CM becomes a relay node to provide the required interconnections to the child clusters (through a TM with lower HC) and/or MPs/sink (via a TM with higher HC). Otherwise, a FRR message is sent until a RREP message is received from the available TMs.

4. **A TM failure:** TM is a node which resides on the tree backbone and receives both RREQ and RREP during the backbone establishment phase (for example, node 31 and 11 in Figure 4-3). The failing TM propagates a TM_{Fail} message to find any other available TM which has already received the same RREQ and RREP. A new TM is found if it has a record of both RREQ and RREP respectively with the same source IDs (srcId). Otherwise, two interconnected nodes (each node has a same record of RREP or RREQ) is selected as the new TMs to cover the disconnected nodes around the failing TM. In fact, the TM node with the same record of RREP is responsible for interconnecting the TM which have the lower HC value, while the TM node with the same RREQ is in charge of interconnecting the TM with the higher HC value. FRR messages are forwarded from the failing TM to find the new TM(s). After receiving the reply of FRR message(s), the failing TM informs the new TM(s) to start their duties by sending TM_{BYE}.

5. **A MP failure:** MP is a TM which has multiple connections to CHs (for example, node 39 in Figure 4-3). Similar to TM failure, a failing MP needs to find the nodes (new MPs) which have the potential to cover the disconnections caused by the failure. The number of new MPs varies depending on the number of connected CHs to the MPs. The failing MP needs to cover failing links between TMs which have lower and higher HC values. In this case, it propagates a MP_{BYE} message to its single-hop neighbours which do not have a lower HC value. Each node which receives the message sends a FRR message to join the tree. The FRR messages are forwarded to neighbour with a fewer HC value until they are received by

active TMs. TM nodes which receive multiple FRR from different nodes become new MPs. The new MPs reply back the FRR request using the RREP messages to start their duties.

The performance of ZMA and CBA needs to be extended by considering unexpected node failures during the data aggregation procedure. As nodes fail suddenly, there is no way to inform the neighbour nodes in advance of the failure. Besides, wireless sensor nodes usually utilise a connection-less model of communication to transmit the network packets. Hence, a sender node never knows about a failure in its neighbourhood as no acknowledgement message is supposed to be received. Utilising a connection-based MAC protocols such as CSMA/CA [KredoII and Mohapatra, 2007] would be a potential solution to resolve this drawback. The connection-based MAC protocols let the sender nodes check the availability of destination node in advance of transmitting packets. Thus, a data packet is transmitted only if the sender ensures that there is still a reliable link to the receiver. However, connection-based MAC protocols are extremely costly (in terms of energy) in WSNs. They would increase the network energy consumption due to a greater number of transmitted control packets (i.e RTS/CTS), especially when the network is large and dense. Occasional routing infrastructure reconstruction also can be a potential solution to deal with unexpected node failures. The sink asks the sensor node to reconstruct the routing infrastructure (i.e tree, clusters and/or zones) at a set of specific periods. The reconstruction time can be calculated according to the average working time of sensor nodes and/or failure probability depending on the environment safety (node capture attacks) and/or node hardware characteristics (hardware damage). The time values and infrastructure reconstruction frequency are assigned to the cost/zone messages which are forwarded from the sink during network deployment phase. Hence, the nodes would set a timer to perform the routing infrastructure establishment algorithm occasionally.

Extending CBA and ZMA to support mobile sensor nodes (MWSN) is an issue that needs to be addressed as future work. In MWSN, the establishment/maintenance cost of routing infrastructure (the spanning tree in CBA or data regions in ZMA) is increased due to the frequent network topology changes caused by node mobility. Social networking [Scott, 2000] is a potential technique to reduce the update cost of routing infrastructure according to the topology changes, especially when the nodes are mobile and/or the network is highly dynamic [Dinh et al., 2009], [Aggarwal and Abdelzaher, 2011]. Social networking patterns such as content-based relations (or common interest relationships) [Daly and Haahr, 2007] can be used by the disconnected nodes (caused by mobility and/or topology change) to (re-)join the routing infrastructure (i.e spanning tree). Using this technique, the disconnected nodes would firstly try to contact sensor nodes which have a better communication history in terms of frequency and/or duration with the nodes residing on the infrastructure (TMs and/or ZMACs). This would increase the probability of relaying data messages from the disconnected nodes to the available nodes which reside on the routing infrastructure, according to the communication histories which show previous an/or potential connections. Hence, the disconnected nodes join the routing infrastructure more quickly and by transmitting a fewer number of control packets. For this reason, utilising social networking pattern would reduce the routing infrastructure maintenance energy cost and or delay in CBA and/or ZMA, regardless of node location or mobility pattern.

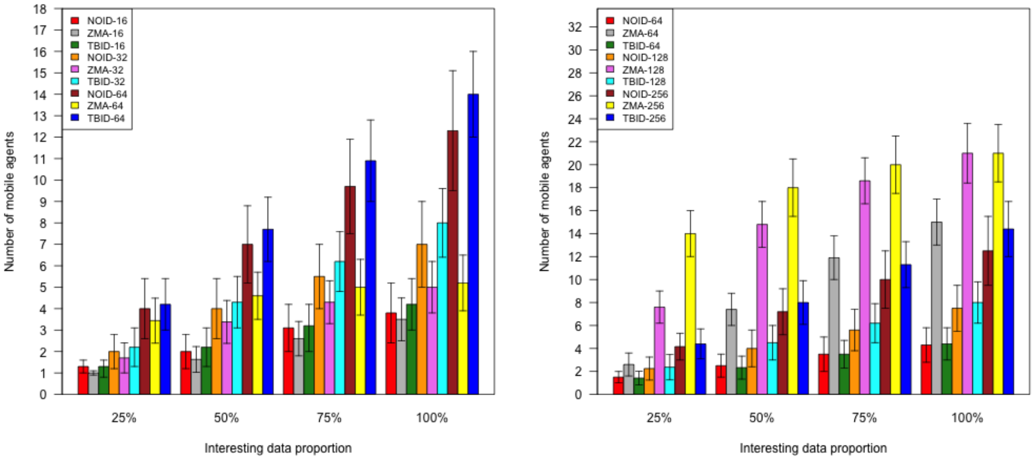
6.3.4 Delay-Accuracy correlation: Further Investigations

Further investigations into the correlation of accuracy and delay needs to be considered as part of future work. Considering the delay-accuracy correlation lets researchers and engineers design data aggregation routing protocols that have the potential to balance the accuracy and delay according to the data consumer requirements. Hence, it offers benefits to WSN applications which are highly time sensitive. According to the results in Section 5.3, delay (ETE) is not directly correlated to accuracy. This is because of the impact of additional parameters such as environmental noise (jamming attack), the degree of parallelism in data aggregation (the number of active MAs), communication patterns (single/multi-hop) and/or network traffic (access time). Investigating of the impact of each parameter on delay may lead to an analytical model that shows how delay changes according to increasing amounts of captured/delivered data during data aggregation routing.

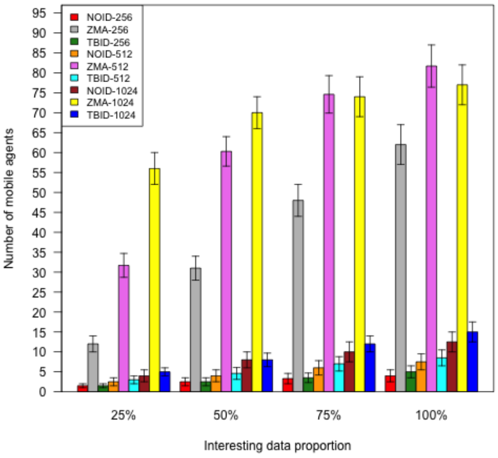
Appendix A

The results of data aggregation routing protocols

Figure A-1: Number of mobile agent in MA routing protocols.

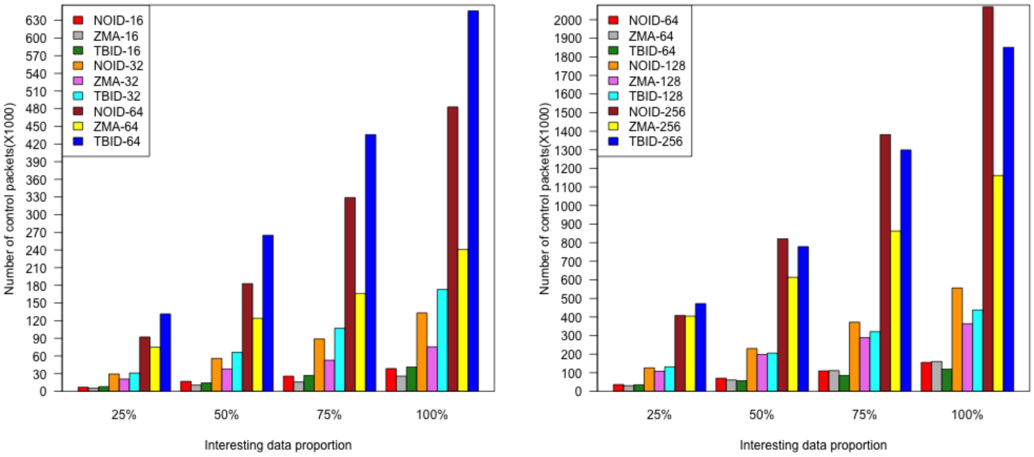


(a) small area (200X200)m^2 (b) medium area (400X400)m^2

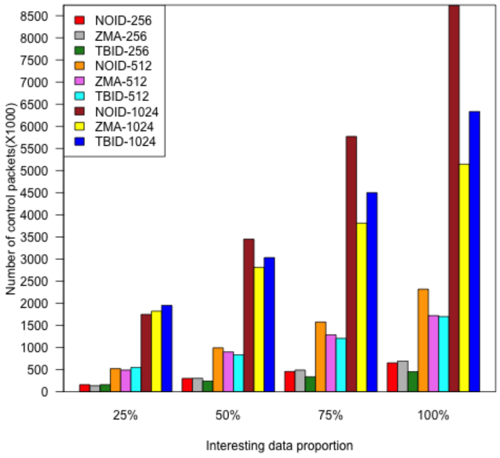


(c) large area (800X800)m^2

Figure A-2: Transmitted control packets in MA routing protocols.

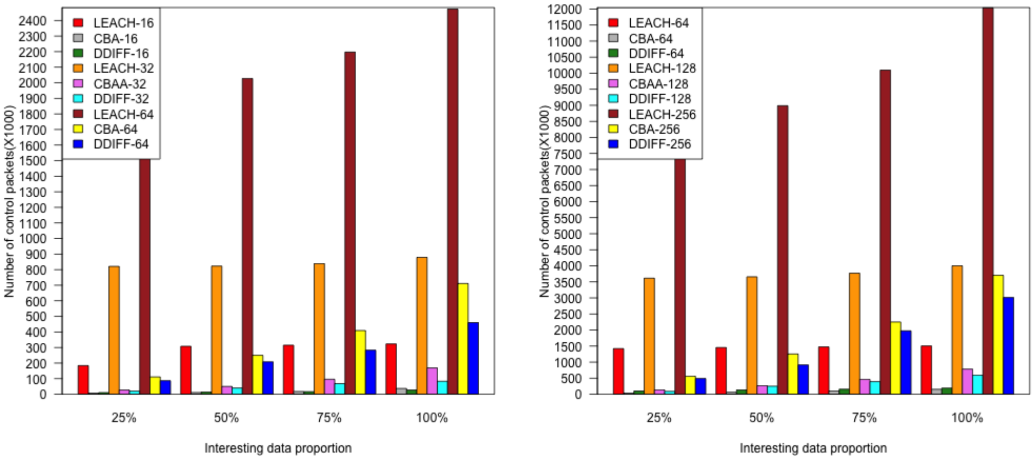


(a) small area (200X200)m^2 (b) medium area (400X400)m^2



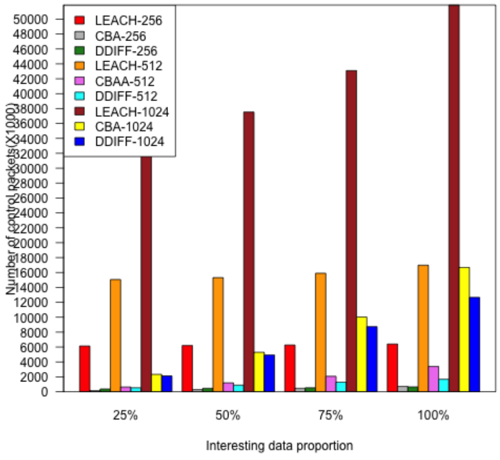
(c) large area (800X800)m^2

Figure A-3: Transmitted control packets in client/server routing protocols.



(a) small area (200X200)m^2

(b) medium area (400X400)m^2



(c) large area (800X800)m^2

Appendix B

Simulation Setup Files

B.1 Simulation Setup File (.ini) for ZMA

This is the input file for the simulator MiXiM [Viklund, 2013], corresponding to the experiment with parameters: protocol name= ZMA, node count= 256, field size= $800 \times 800m^2$, data density= 100%, described on page 100.

```
[General]
network = ZMA # protocol name#
sim-time-limit = 3600s # simulation time #
cmdenv-express-mode = true
cmdenv-module-messages = false
repeat = 300 # simulation repetition#

[Simulation parameters]
**.playgroundSizeX = 800m #Area Size #
**.playgroundSizeY = 800m
**.playgroundSizeZ = 800m # This is not used as the experiments are set up in a 2D field#
**.numNodes = 256 # Node Count #
**.world.use2D = true #two Dimensions#

[Parameters for the Connection Manager]
**.connectionManager.carrierFrequency = 2.412e9Hz
# max transmission power [mW] #
**.connectionManager.pMax = 0.4mW
```

```
# signal attenuation threshold [dBm] #
**.connectionManager.sat = -100dBm
# path loss coefficient alpha #
**.connectionManager.alpha = 3.0
**.connectionManager.sendDirect = false
```

[**Parameters for the Host (NIC)**]

```
**.nic.sleepCurrent = 0.02mA
**.nic.rxCurrent = 16.4mA
**.nic.decodingCurrentDelta = 0mA
**.nic.txCurrent = 17mA
**.nic.setupRxCurrent = 8.2mA
**.nic.setupTxCurrent = 8.2mA
**.nic.rxTxCurrent = 17mA
**.nic.txRxCurrent = 17mA
```

[**Physical layer parameters**]

```
**.phy.usePropagationDelay = false
**.phy.thermalNoise = -110dBm
**.phy.useThermalNoise = true
**.phy.analogueModels = xmldoc(”config.xml”)
**.phy.decider = xmldoc(”config.xml”)
**.phy.maxTXPower = 110.11mW
**.phy.sensitivity = -119.5dBm
**.phy.initialRadioState = 0
```

[**MAC layer parameters**]

```
**.node[*].nic.mac.animation = true
**.node[*].nic.mac.debug = true
**.node[*].nic.mac.queueLength = 5
**.node[*].nic.mac.headerLength = 40bit
**.node[*].nic.mac.bitrate = 15360bps
**.node[*].nic.mac.txPower = 100mW
**.node[*].nic.mac.stats = true
**.node[*].nic.mac.useMACAcks = true
**.node[*].nic.mac.checkInterval = 0.1s
**.node[*].nic.mac.slotDuration = 1s
```

```
    [Battery parameters]
**.battery.nominal = 99999mAh
**.battery.capacity = 99999mAh
**.battery.voltage = 3.3V
**.battery.resolution = 10s
**.battery.publishDelta = 0.1
**.battery.publishTime = 0
**.battery.numDevices = 1

    [Application layer parameters]
**.node[*].applicationType = ”SensorApplLayer”
**.appl.trafficType = ”periodic”
**.appl.initializationTime = 10s
**.node[0].appl.nbPackets = 0
**.node[1..255].appl.nbPackets = 1
**.node[1..255].netwl.data = true
**.node[*].appl.destAddr = 0
**.node[*].appl.trafficParam = 30 s

    [NETW layer parameters]
**.node[*].networkType = ”ZMA”
**.node[*].netwl.trace = true
**.node[*].netwl.stats = true
**.node[*].netwl.headerLength = 40 bit
**.node[*].netwl.sinkAddress = 0
**.node[0].netwl.routeFloodsInterval = 0 s
#Forming the data regions#
**.node[*].netwl.DataInter = 1000s
#MA fire time#
**.netwl.MAInter = 2500s
# RSSI threshold for route selection #
**.node[*].netwl.rssiThreshold = -50 dBm

    [Field parameters]
**.node[*].mobilityType = ”StationaryMobility”
**.node[0].mobility.initialX = 400m
```

```
**.node[0].mobility.initialY = 400m
**.node[0].mobility.initialZ = 400m
```

B.2 Simulation Setup File (.ini) for CBA

This is the input file for the simulator MiXiM [Viklund, 2013], corresponding to the experiment with parameters: protocol name= CBA, node count= 256, field size= $800 \times 800m^2$, data density= 100%, described on page 100.

```
   [General]
network = CBA # protocol name#
sim-time-limit = 3600s # simulation time #
cmdenv-express-mode = true
cmdenv-module-messages = false
repeat = 300 # simulation repetition#

   [Simulation parameters]
**.playgroundSizeX = 800m #Area Size #
**.playgroundSizeY = 800m
**.playgroundSizeZ = 800m # This is not used as the experiments are set up in a 2D field#
**.numNodes = 256 # Node Count #
**.world.use2D = true #two Dimensions#

   [Parameters for the Connection Manager]
**.connectionManager.carrierFrequency = 2.412e9Hz
# max transmission power [mW] #
**.connectionManager.pMax = 0.4mW
# signal attenuation threshold [dBm] #
**.connectionManager.sat = -100dBm
# path loss coefficient alpha #
**.connectionManager.alpha = 3.0
**.connectionManager.sendDirect = false

   [Parameters for the Host (NIC)]
**.nic.sleepCurrent = 0.02mA
**.nic.rxCurrent = 16.4mA
**.nic.decodingCurrentDelta = 0mA
**.nic.txCurrent = 17mA
**.nic.setupRxCurrent = 8.2mA
```

```
**.nic.setupTxCurrent = 8.2mA
**.nic.rxTxCurrent = 17mA
**.nic.txRxCurrent = 17mA

    [Physical layer parameters]
**.phy.usePropagationDelay = false
**.phy.thermalNoise = -110dBm
**.phy.useThermalNoise = true
**.phy.analogueModels = xmldoc(”config.xml”)
**.phy.decider = xmldoc(”config.xml”)
**.phy.maxTXPower = 110.11mW
**.phy.sensitivity = -119.5dBm
**.phy.initialRadioState = 0

    [MAC layer parameters]
**.node[*].nic.mac.animation = true
**.node[*].nic.mac.debug = true
**.node[*].nic.mac.queueLength = 20
**.node[*].nic.mac.headerLength = 40bit
**.node[*].nic.mac.bitrate = 15360bps
**.node[*].nic.mac.txPower = 100mW
**.node[*].nic.mac.stats = true
**.node[*].nic.mac.useMACAcks = true
**.node[*].nic.mac.checkInterval = 0.1s
**.node[*].nic.mac.slotDuration = 1s

    [Battery parameters]
**.battery.nominal = 99999mAh
**.battery.capacity = 99999mAh
**.battery.voltage = 3.3V
**.battery.resolution = 10s
**.battery.publishDelta = 0.1
**.battery.publishTime = 0
**.battery.numDevices = 1

    [Application layer parameters]
**.node[*].applicationType = ”SensorApplLayer”
```

```
**.appl.trafficType = ”periodic”
**.appl.initializationTime = 10s
**.node[0].appl.nbPackets = 0
**.node[1..255].appl.nbPackets = 1
**.node[1..255].netwl.data = true
**.node[*].appl.destAddr = 0
**.node[*].appl.trafficParam = 30 s

   [NETW layer parameters]
**.node[*].networkType = ”CBA”
**.node[*].netwl.trace = true
**.node[*].netwl.stats = true
**.node[*].netwl.headerLength = 40 bit
**.node[*].netwl.sinkAddress = 0
# Cost allocations time #
**.node[0].netwl.routeFloodsInterval = 1000 s
# Vicinity discovery interval for tree establishment #
**.netwl.Vic-Inter = 200 s
# Clustering time#
**.node[*].netwl.DataInter = 1500 s
# Data Aggregation Fire Time#
**.netwl.CHFire = 2500 s
# RSSI threshold for route selection #
**.node[*].netwl.rssiThreshold = -50 dBm

   [Field parameters]
**.node[*].mobilityType = ”StationaryMobility”
**.node[0].mobility.initialX = 400m
**.node[0].mobility.initialY = 400m
**.node[0].mobility.initialZ = 400m
```

Bibliography

Ameer Ahmed Abbasi and Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30:28262841, 2007. 131

Federal Emergency Management Agency. Emergency alert system, 2013. `http://transition.fcc.gov/pshs/services/eas/`, Retrieved (December, 2013). 5

Charu C. Aggarwal and Tarek F. Abdelzaher. *Social Network Data Analytics*, chapter Integrating Sensors and Social Networks, pages 379–412. Springer, 2011. 215

Francesco Aiello, Giancarlo Fortino, Antonio Guerrieri, and Raffaele Gravina. Maps: A mobile agent platform for wsns based on java sun spots. *A workshop of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-09), Budapest, Hungary, 12 May*, pages 25–32, 2009. 69

Kemal Akkaya and Mohamed Younis. An energy aware qos routing protocol for wireless sensor networks. *23rd International Conference on Distributed Computing Systems (ICDCS 2003), Providence, RI, USA, May 19-22*, pages 710–715, 2003. 61

Kemal Akkaya and Mohamed F. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005. 17, 41, 42, 43, 45

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

Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: A survey. *Ieee Wireless Communications*, 11(6):6–28, 2004. 17, 41, 42, 43, 65

Hitha Alex, Mohan Kumar, and Behrooz Shirazi. Midfusion: An adaptive middleware for information fusion in sensor network applications. *Information Fusion*, 9(3): 332–343, 2008. 66

Muneeb Ali, Umar Saif, Adam Dunkels, Thiemo Voigt, Kay Romer, Koen Langendoen, Joseph Polastre, and Zartash Afzal Uzmi. Medium access control issues in sensor networks. *ACM SIGCOMM Computer Communication Review*, 36(2):33–36, 2006. 132

Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009. 133

Mohammad Hossein Anisi, Abdul Hanan Abdullah, and Shukor Abd Razak. Efficient data aggregation in wireless sensor networks. *International Conference on Future Information Technology (ICFIT'11), Singapore, Singapore*, 13:305–310, September 16-18, 2011. 66

Arvind Arasu, Venkatesh Ganti, and Raghav Kaushik. Efficient exact set-similarity joins. *the 32Nd International Conference on Very Large Data Bases (VLDB'06), Seoul, Korea, september 12-15*, pages 918–929, 2006. 132

Saeid Pourroostaei Ardakani, Julian Padget, and Marina De Vos. Hrts: A hierarchical reactive time synchronization protocol for wireless sensor networks. *Ad Hoc Networks*, 129:47–62, 2014. 52, 89, 114, 157, 211

B. Zurita Ares, C. Fischione, and K. H. Johansson. *Wireless Sensor Networks*, volume 4373 of *Lecture Notes in Computer Science*, chapter Energy consumption of minimum energy coding in CDMA wireless sensor networks, pages 212–227. Springer Berlin/Heidelberg, Delft, The Netherlands, 2007. 52

Miguel Baguena, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. Towards realistic vehicular network simulation models. *Wireless Days 2012 (WD'12), Dublin, Ireland, November 21-23*, pages 1–3, 2012. 101

Humayun Bakht. Survey of routing protocols for mobile ad-hoc network. *International Journal of Information and Communication Technology Research*, 1 (6):258–270, 2011. 27, 32

Shadi Saleh Ali Basurra. *Collision Guided Routing for Ad-Hoc Mobile Wireless Networks.* PhD thesis, Department of Computer Science, University of Bath, October 2012. 20, 21, 22, 107, 140, 169

Richard E. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16 (1958):87–90, 1956. 23, 24, 36

Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. *The 3rd ACM International Symposium on Mobile Ad Hoc Networking (MobiHoc '02), Lausanne, Switzerland, June 9-11*, pages 80–91, 2002. 104

Antonia Bezenchek, Maurizio Rafanelli, and Leonardo Tininini. A data structure for representing aggregate data. *the Eighth International Conference on Scientific and Statistical Database Management*, pages 22–31, 1996. 11

Andrzej Bieszczad, Bernard Pagurek, and Tony White. Mobile agents for network management. *IEEE Communications Surveys*, 1(1):2–9, 1998. 69

Pratik K. Biswas, Hairong Qi, and Yingyue Xu. Mobile-agent-based collaborative sensor fusion. *Information Fusion*, 9(3):399–411, 2008. 12, 63

Philippe Bonnet, Johannes Gehrke, and Praveen Seshadri. Towards sensor database systems. *Second International Conference on Mobile Data Management (MDM '01), Hong Kong, China, January 8-10*, pages 3–14, 2001. 42

JAVIER BONNY. Investigating mac power consumption in wireless sensor network. Technical report, Department of Communication Systems, Swiss Institute of Technology (EPFL), 2005. 171

Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z. Ahmad, Ladislau Blni, and Damla Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55:30323080, 2011. 21, 28, 32, 35

Athanassios Boulis, Saurabh Ganeriwal, and Mani B. Srivastava. Aggregation in sensor networks: an energyaccuracy trade-off. *Ad Hoc Networks*, 1(317-331), 2003. 99, 106, 168, 173, 176, 177, 178, 194

David Braginsky and Deborah Estrin. Rumor routing algorithm for sensor networks. *The first ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02), Atlanta, Georgia, USA, September 28*, pages 22–31, 2002. 50

Josh Broch, David A. Maltzand David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. *The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom98), Dallas, Texas, USA, October 2530*, pages 85–97, 1998. 87

Wei Cai, Min Chen, Takahiro Hara, and Lei Shu. Ga-mip: Genetic algorithm based multiple mobile agents itinerary planning in wireless sensor networks. *The 5th Annual*

International Wireless Internet Conference, Singapore, March 1-3, pages 1–8, 2010. 74

Yongtao Cao, Chen He, and Lingge Jiang. Energy-efficient routing for mobile agents in wireless sensor networks. *Frontiers of Electrical and Electronic Engineering in China*, 2(2):161–166, 2007. 69

Ian D. Chakeres and Elizabeth M. Belding-Royer. The utility of hello messages for determining link connectivity. *5th International Symposium on Wireless Personal Multimedia Communications (WPMC), Honolulu, Hawaii, October 27-30*, 2:504–508, 2002. 87, 89

Min Chen, Taekyoung Kwon, Yong Yuan, and Victor C.M. Leung. Mobile agent based wireless sensor networks. *Journal of computers*, 1(1):14–21, April 2006. 72

Min Chen, Taekyoung Kwon, Yong Yuan, Yanghee Choi, and Victor C. M. Leung. Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007:1–13, 2007. 71, 72, 83, 106

Min Chen, Sergio Gonzlez, Yan Zhang, and Victor C.M. Leung. Multi-agent itinerary planning for sensor networks. *Quality of Service in Heterogeneous Networks*, 22: 584–597, 2009a. 99, 104, 106

Min Chen, Victor Leung, Shiwen Mao, Taekyoung Kwon, and Ming L. Energy-efficient itinerary planning for mobile agents in wireless sensor networks. *IEEE International Conference on Communications (ICC), Dresden, Germany, June 14-18*, pages 1–5, 2009b. 73, 106

Maurice Chu, Horst Haussecker, and Feng Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16:293–313, 2002. 59

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT press, 1990. 23, 36

Crossbow. Telosb mote platform, 2014. `http://www.willow.co.uk/TelosB_Datasheet.pdf`, Reterived (December 2014). 101

Elizabeth Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. *the 8th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc 07, Montreal, Quebec, Canada, September 9-14*, pages 32–40, 2007. 215

Koustuv Dasgupta, Konstantinos Kalpakis, and Parag Namjoshi. An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. *IEEE Wireless Communications and Networking, WCNC 2003,New Orleans, LA, USA, 16-20 March*, pages 1948–1953, 2003. 68, 131

Edison Pignaton de Freitas. *Cooperative Context-Aware Setup and Performance of Surveillance Missions Using Static and Mobile Wireless Sensor Networks*. PhD thesis, Halmstad University, 2011. 69

Ilker Demirkol, Cem Ersoy, and Fatih Alagoz. Mac protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006. 132, 170

J. Deng, Y. S. Han, P. N. Chen, and P. K. Varshney. Optimal transmission range for wireless ad hoc networks based on energy efficiency. *IEEE Trans. on Communications*, 55(9):1772–1782, September 2007. 86

Tang N. Dinh, Ying Xuan, and My T. Thi. Towards social-aware routing in dynamic communication networks. *28th International Performance Computing and Communications Conference, IPCCC 2009, , Phoenix, Arizona, USA, 14-16 December*, pages 161–168, 2009. 215

Rohit Dube, Cynthia D. Rais, Kuang yeh Wang, and Satish K. Tripathi. Signal stability based adaptive routing (ssa) for ad-hoc mobile networks. *IEEE Personal Communications Magazine*, 4:36–45, 1997. 27

Muhamnmad Omer Farooq, Abdul Basit Dogar, and Ghalib Asadullah Shah. Mr-leach: Multi-hop routing with low energy adaptive clustering hierarchy. *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010), Venice/Mestre, Italy, July 18 - 25*, pages 262–68, 2010. 52

Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: A survey. *Wireless Communications*, 14(2): 70–87, 2007. 11

J. J Garcia-Luna-Aceves and Marcelo Spohn. Source-tree routing in wireless networks. *Seventh Annual International Conference on Network Protocols (ICNP '99), Toronto, Ontario, Canada, October 31-November 3*, pages 273–82, 1999. 24

Leszek Gasieniec, Jesper Jansson, and Andrzej Lingas. Approximation algorithms for hamming clustering problems. *Journal of Discrete Algorithms*, 2:289–301, 2004. 132

Damianos Gavalas, Aristides Mpitziopoulos, Grammati Pantziou, and Charalampos Konstantopoulos. An approach for near-optimal distributed data fusion in wireless sensor networks. *Wireless Networks*, 16:1407–1425, 2010. 14, 15, 74, 82, 84, 92, 96, 99, 102, 107, 126, 178, 201

Pravin Ghosekar, Girish Katkar, and Dr. Pradip Ghorpade. Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, 2003. 18

Inc Harbor Research. The 2013 wireless sensor networks report, 2013. `http://www.researchandmarkets.com/research/lrpv7z/the_2013_wireless`, Retrieved (December, 2013). 2

Peter Harrop and Raghu Das. The new market for ubiquitous sensor networks (usn), 2013. `http://www.idtechex.com/research/reports/wireless-sensor-networks-2011-2021-000275.asp`, Retrieved (December, 2013). 2

Christopher Haslett. *Essentials of radio wave propagation*. Cambridge University Press, 2008. 91

Tian He, John A Stankovic, Chenyang Lu, and Tarek Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. *The 23rd International Conference on Distributed Computing Systems (ICDCS' 03), Providence, RI, USA, May 19-22*, pages 46–55, 2003. 60

Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An application specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–70, 2002. 53, 130

Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *The 33rd Hawaii International Conference on System Sciences (HICSS'00), the Island of Maui, 4-7 January*, pages 3005–3014, 2000. 14, 15, 52, 91, 92, 128, 129, 130, 131, 145, 148, 184, 203

William D Henderson and Steven Tron. Verification of the minimum cost forwarding protocol for wireless sensor networks. *11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Prague, Czech Republic, September 20-22*, pages 194–201, 2006. 49

Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt, and Piet Demeester. An overview of mobile ad hoc networks:applications and challenges. Technical report, Department of Information Technology (INTEC), Ghent University, 2004. 18

Barbara Hohlt, Lance Doherty, and Eric Brewer. Flexible power scheduling for sensor networks. *The 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04), Berkeley, California, USA, April 26-27*, pages 205–214, 2004. 170

Fei Hu, Xiaojun Cao, and Carter May. Optimized scheduling for data aggregation in wireless sensor networks. *International Symposium on Information Technology: Coding and Computing (ITCC 2005), Las Vegas, Nevada, USA, April 4-6*, 2:557–561, 2005. 65

Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *The 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00), Boston, Massachusetts, August 6-11*, pages 56–67, 2000. 14, 15, 46, 47, 62, 64, 87, 128, 129, 145, 148, 184, 203

Chalermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. Impact of network density on data aggregation in wireless sensor networks. *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pages 457–459, 2002. 11

P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. *IEEE International Multi Topic Conference 2001 (IEEE INMIC 2001), 28-30 December*, pages 62–28, 2001. 23

Congfeng Jiang, Daomin Yuan, and Yinghui Zhao. Towards clustering algorithms in wireless sensor networks-a survey. *IEEE Conference on Wireless Communications and Networking Conference (WCNC'09), Budapest, Hungary, April 5-8*, pages 2009–2014, 2009. 131

Mario Joa-Ng and I-Tai Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 17(8):1415–1425, 1999. 30

David B. Johnson, David A. Maltz, and Josh Broch. *Ad Hoc Networking*, chapter DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesle, 2001. 26, 34

Joseph Kabara and Maria Calle. Mac protocols used by wireless sensor networks and a general method of performance evaluation. *International Journal of Distributed Sensor Networks*, 2012:1–11, 2012. 101

Vikas Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. *The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, California, USA, March 30- April 3*, pages 459–469, 2003. 36

Asif Khan, Israfil Tamim, Emdad Ahmed, and Muhammad Abdul Awal. Multiple parameter based clustering (mpc): Prospective analysis for effective clustering in wireless sensor network (wsn) using k-means algorithm. *Wireless Sensor Network*, 4:18–24, 2012. 91

Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, July 2000. ISSN 1022-0038. doi: 10.1023/A:1019106118419. URL `http://dx.doi.org/10.1023/A:1019106118419`. 32

Charalampos Konstantopoulos, Aristides Mpitziopoulos, Damianos Gavalas, and Grammati Pantziou. Effective determination of mobile agent itineraries for data aggregation on sensor networks. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 22(12):1679–1693, 2010. 14, 15, 75, 82, 84, 87, 96, 99, 102, 107, 126, 178, 201

Kurtis B. KredoII and Prasant Mohapatra. Medium access control in wireless sensor networks. *Computer Networks*, 51(4):961–994, 2007. doi: 10.1016/j.comnet.2006.06.012. URL `http://dx.doi.org/10.1016/j.comnet.2006.06.012`. 215

Bhaskar Krishnamachari and John Heidemann. Application specific modeling of information routing in wireless sensor networks. *In the prceeding of IEEE international performance, computing and communications conference,*, 23:717–722, 2004. 64

Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker. Modelling data-centric routing in wireless sensor networks. *The 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, USA, June 23-27*, 2(4):1–11, 2002. 9, 13, 37, 85, 133, 134

Joanna Kulik, Wendi Rabiner, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *The 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom' 99), Seattle, Washington, August 15-20*, pages 174–185, 1999. 46, 62, 64

A. Kpke, M. Swigulski, K. Wessel, D. Willkomm, P.T. Klein Haneveld, T.E.V. Parker, O.W. Visser, H.S. Lichte, and S. Valenti. Simulating wireless and mobile networks in omnet++ the mixim vision. *the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (Simutools '08), Marseille, France, March 3-7*, pages 71–78, 2008. 99

Harvard Sensor Networks Lab. Codeblue: Wireless sensors for mdical care, 2013. http://fiji.eecs.harvard.edu/CodeBlue, Retrieved (December, 2013). 5

Olaf Landsiedel, Klaus Wehrle, and Stefan Gotz. Accurate prediction of power consumption in sensor networks. *the 2Nd IEEE Workshop on Embedded Networked Sensors(EmNets '05), Sydney, Australia, May 30-31*, pages 37–44, 2005. 169

Danny B. Lange, Mitsuru Oshima, Gnter Karjoth, and Kazuya Kosaka. Aglets: Programming mobile agents in java. *Worldwide Computing and Its Applications (WWCA97), Lecture Notes in Computer Science*, 1274, 1997. 69

Sung-Ju Lee and Mario Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. *The IEEE International Conference on Communications (ICC' 2001), Helsinki, Finland, June 11-14*, 10:3201–3205, 2001. 34

Sergio Gonza Lez-Valenzuela, Min Chen, and Victor C.M Leung. Applications of mobile agent in wireless networks and mobile computing. *Advances in Computers*, 82:113–163, 2011. 69

Stephanie Lindsey, Cauligi S. Raghavendra, and Krishna M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel Distributed System*, 13(9):924–935, 2002. 53, 54, 68, 168, 176, 178, 194

Changling Liu and Jorg Kaiser. A survey of mobile ad hoc network routing protocols. Technical report, University of Ulm, 2005. 22, 23, 28

Kai-Wei Fan Sha Liu and Prasun Sinha. Structure-free data aggregation in sensor networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 6(8):929–942, August 2007. 68

Xuxun Liu. A survey on clustering routing protocols in wireless sensor networks. *Sensors*, 12(11113-11153), 2012. 130, 136

V. Loscr, G. Morabito, and S. Marano. A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach). *62nd Vehicular Technology Conference (VTC' 2005-fall), September 25-28*, pages 1809–13, 2005. 52

Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: A tiny aggregation service for ad-hoc sensor networks. *Fifth Symposium on Operating Systems Design and implementation (OSDI 02), Boston, MA, USA, December 9 - 11*, pages 131–146, 2002. 66

Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02), Atlanta, Georgia, USA, September 28 - 28*, pages 88–97, 2002. 5

Arati Manjeshwar and Dharma P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. *15th International Parallel and Distributed Processing Symposium (IPDPS), San Francisco, USA, April 23-27*, 3:2009 – 2015, 2001. 54, 131, 144

Arati Manjeshwar and Dharma P. Agrawal. Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *the 16th International Parallel and Distributed Processing Symposium (IPDPS '02), Florida, USA, April 15-19*, pages 195–202, 2002. 55, 144

Jianlin Mao, Zhiming Wua, and Xing Wuc. A tdma scheduling scheme for many-to-one communications in wireless sensor networks. *Computer Communications*, 30(4): 863–872, February 2007. 52

Miklos Maroti, Branislav Kusy, Gyula Simon, and Akos Ledeczi. The flooding time synchronization protocol. *the 2Nd International Conference on Embedded Networked Sensor Systems (sensys'04), Baltimore, MD, USA, November 3-5*, pages 39–49, 2004. 173

Evangelos E. Milios. An object-oriented signal processing environment: The knowledge-based signal processing package. Technical report, Massachusetts Institute of Technology, 1984. 3

Matthew J. Miller and Nitin H. Vaidya. A mac protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 4(3):228–242, June 2005. 171

Aravind B Mohanoor, Sridhar Radhakrishnan, and Venkatesh Sarangan. Online energy aware routing in wireless networks. *Ad Hoc Networks*, 7(5):918–931, July 2009. ISSN 1570-8705. doi: 10.1016/j.adhoc.2008.08.004. URL `http://dx.doi.org/10.1016/j.adhoc.2008.08.004`. 35

Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. *Mobile Agent Middleware for Autonomic Data Fusion in Wireless Sensor Networks*, chapter Autonomic Computing and Networking, pages 57–81. Springer, 2009. 70, 71, 73

F. NADEEM, S. CHESSA, E. LEITGEB, and S. ZAMAN. The effects of weather on the life time of wireless sensor networks using fso/rf communication. *Radioengineering Journal*, 19(2):262–270, June 2010. 101

Amit N.Thakare and Mrs. M. Y. Joshi. Performance analysis of aodv and dsr routing protocol in mobile ad hoc networks. *IJCA Special Issue on MANETs*, 4:211–218, 2010. Published by Foundation of Computer Science. 27

Hyacinth S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(3)(205-44), October/November 1996. 69

OMNET++. Omnet++ simulator, 2012. `http://www.omnetpp.org/`, Retrieved (March 2012). 14, 99, 145

Guangyu Pei, Mario Gerla, and Xiaoyan Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility. *First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC. 2000)*, pages 11–18, 2000. 29, 30

Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM*, 24(4):234–244, 1994. 24, 36

Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. *Second IEEE Workshop on Mobile Computer Systems and Applications (WMCSA '99), New Orleans, Louisiana, USA, February 25-26*, pages 90–100, 1999. 26, 34, 89

Hairong Qi and Feiyi Wang. Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. *In the proceedings of the13th International Conference on Wireless Communication, Calgary, Alberta, Canada, July 9-11*, pages 147–153, 2001. 73

Hairong Qi, S. Sitharama Iyengar, and Krishnendu Chakrabarty. Multiresolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(3):383–91, 2001. 69, 70, 72

Hairong Qi, Yingyue Xu, and Xiaoling Wang. Mobile-agent-based collaborative signal and information processing in sensor networks. *Proceedings of the IEEE*, 91(8):1172–83, 2003. 72

Ramesh Rajagopalan and Pramod K. Varshney. Data aggregation techniques in sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 8:48–63, 2006. 11, 64, 67

Venugopalan Ramasubramanian, Ranveer Chandra, and Daniel Mosse. Providing a bidirectional abstraction for unidirectional ad hoc networks. *The 21st Annual Joint Conference of the IEEE Computer and Communications Societies, New York, USA, June 23-27*, pages 345–354, 2002. 142

Ge Ran, Huazhong Zhang, and Shulan Gong. Improving on leach protocol of wireless sensor networks using fuzzy logic. *Journal of Information & Computational Science*, 7(3):767775, 2010. 53

Richard F. Rashid and George G. Robertson. Accent: A communication oriented network operating system kernel. *Proceedings of the Eighth ACM Symposium on Operating Systems Principles (SIGOPS Operating System Review)*, 15:64–75, 1981. ISSN 0163-5980. 3

Marcelo G. Rubinstein, Igor M. Moraes, Miguel Elias M. Campista, Lu Henrique M. K. Costa, and Otto Carlos M. B. Duarte. A survey on wireless ad hoc networks. *Mobile and Wireless Communication Networks, Santiago, Chile, August 20-25*, pages 1–33, 2006. 18, 22

Narayanan Sadagopan, Bhaskar Krishnamachari, and Ahmed Helmy. Active query forwarding in sensor networks (acquire). *Ad Hoc Networks*, 3:91–113, 2005. 10, 59

Prince Samar, Marc R. Pearlman, and Zygmunt J. Haas. Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks. *ACM Trans. Netw.*, 12(4):595–608, August 2004. ISSN 1063-6692. doi: 10.1109/TNET.2004.833153. URL `http://dx.doi.org/10.1109/TNET.2004.833153`. 29

Curt Schurgers and Mani B. Srivastava. Energy efficient routing in wireless sensor networks. *IEEE Military Communications Conference (MILCOM 2001), Communications for Network-Centric Operations: Creating the Information Force, Tysons Corner McLean, Virginia USA, 28-31 October*, 1:357 – 361, 2001. 47

John Scott. *Social Network Analysis: a handbook*. SAGE Publications Ltd, 2000. 215

Yu-Chee Seng, Sheng-Po Kuo, Hung-Wei Lee, and Chi-Fu Huang. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. *2nd international conference on Information processing in sensor networks (IPSN'03), Palo Alto, California, USA, April 22-23*, pages 625–641, 2003. 75, 76, 80

Evan Serqeant. Sample size to estimate a single mean with specified precision, 2014. `http://epitools.ausvet.com.au/content.php?page=1Mean&Stdev=45&Conf=0.95&Error=20`, Retrieved (December, 2014). 102

PREETI SETHI, DIMPLE JUNEJA, and NARESH CHAUHAN. A mobile agent-based event driven route discovery protocol in wireless sensor network: Aerdp. *International Journal of Engineering Science and Technology (IJEST)*, 3(12):8422–9, December 2011. 76

Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *IEEE conference on Wireless Communications and Networking Conference (WCNC2002), Orlando, Florida, USA, MArch 17-21*, 1:350–355, 2002. 48, 49, 134

Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, and Panos K. Chrysanthis. Tina: a scheme for temporal coherency-aware in-network aggregation. *Third ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2003, San Diego, California, USA, September 19*, pages 69–76, 2003. 67

Xingfa Shen, Zhi Wang, Peng Jiang, Ruizhong Lin, and Youxian Sun. Connectivity and rssi based localization scheme for wireless sensor networks. *International Conference on Advances in Intelligent Computing (ICIC'05), Hefei, China, August 23-26*, pages 578–587, 2005. 93

SIMENS. Simens hvac, 2013. `http://www.buildingtechnologies.siemens.com/bt/global/en/buildingautomation-hvac/hvac-products/pages/hvac-products.aspx`, Reterived (April, 2014). 6

Kazem Sohraby, Daniel Minoli, and Taien Znati. *Wireless Sensor Network Technology, Protocols and Applications*. John Wiley & Sons, Inc., 2007. 2, 4, 7, 38, 170

Ignacio Solis and Katia Obraczka. The impact of timing in data aggregation for sensor networks. *IEEE International Conference on Communications (ICC), Paris, France, June, 20-24*, 6:3640–3645, 2004. 9, 85, 106, 134

Ignacio Solis and Katia Obraczka. In-network aggregation trade-offs for data collection in wireless sensor networks. *International Journal of Sensor Network (IJSNet)*, 1 (3/4):200–212, 2006. 9, 168, 178, 194

Chao Song, Ming Liu, Jiannong Cao, Yuan Zheng, Haigang Gong, and Guihai Chen. Maximizing network lifetime based on transmission range adjustment in wireless sensor networks. *Computer Communications*, 2009(32):13161325, 2009. 86, 211

S. Sriram, T. Bheemarjuna Reddy, B. S. Manoj, and C. Siva Ram Murthy. Museqor: Multi-path failure-tolerant security-aware qos routing in ad hoc wireless networks. *HIGH PERFORMANCE COMPUTING - HIPC*, 3296:35–74, 2005. 34

Neelam Srivastava. Challenges of next-generation wireless sensor networks and its impact on society. *JOURNAL OF TELECOMMUNICATIONS*, 1(1):128–133, 2010. 41

Traffic.com. Traffic.com, 2013. `http://here.com/traffic`, Retrieved (December, 2013). 6

David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005. 101

P. Uthansakul, M. E. Bialkowski, S. Durrani, K. Bialkowski, and A. Postula. Effect of line of sight propagation on capacity of an indoor mimo system. *IEEE Antennas and Propagation Society International Symposium 2005, 3-8 July, Washington, DC*, pages 707–710, 2005. 91, 146

Sudarshan Vasudevan, Brian DeCleene, Neil Immerman, Jim Kurose, and Don Towsley. Leader election algorithms for wireless ad hoc networks. *IEEE Computer Society*, pages 261–272, 2003. 84, 89

Andreas Viklund. Mixim code, 2013. `http://mixim.sourceforge.net/index.html`, Retrieved (December, 2013). 99, 100, 145, 146, 221, 225

Meritxell Vinyals, Juan A. Rodriguez-Aguilar, and Jesus Cerquides. A survey on sensor networks from a multiagent perspective. *The Computer Journal*, 54(3):455–470, 2011. 8

Ian Walker. Switching to r: A guide for the behavioural sciences. Technical report, University of Bath, 2013. 102

Nen-Chung Wang, Yung-Fa Huang, and Jhu-Chan Chen. A stable weight-based on-demand routing protocol for mobile ad hoc networks. *Inf. Sci.*, 177(24):5522–5537, 2007a. ISSN 0020-0255. 26

Qin Wang, Mark Hempstead, and Woodward Yang. A realistic power consumption model for wireless sensor network devices. *Third Annual IEEE Communications*

Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Hyatt Regency, Reston, VA, USA, September 25-28, pages 286–295, 2006. 13, 169

Xue Wang, Dao wei Bi, Liang Ding, and Sheng Wang. Agent collaborative target localization and classification in wireless sensor networks. *Sensors*, 7:1359–1386, 2007b. 76

Johanna Williams. Telosb sensor network helpful in many different applications, 2014. `http://telosbsensors.wordpress.com/`, Reterived (April 2014). 38, 126

Qishi Wu, Nageswara S. V. Rao, Jacob Barhen, S. Sitharama Iyengar, Vijay K. Vaishnavi, Hairong Qi, and Krishnendu Chakrabarty. On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 16:740–753, June 2004. ISSN 1041-4347. doi: 10.1109/TKDE.2004.12. URL `http://dl.acm.org/citation.cfm?id=1435713.1437597`. 71, 73, 83, 99, 104, 106

Fan Xiangning and Song Yulin. Improvement on leach protocol of wireless sensor network. *International Conference on Sensor Technologies and Applications (SensorComm 2007), Valencia, Spain, October 14-20*, pages 260–4, 2007. 53

Zixiang Xiong, Angelos D. Liveris, and Samuel Cheng. Distributed source coding for sensor networks. *EEE SIGNAL PROCESSING MAGAZINE*, 21(5):80–94, September 2004. 132

Jiuqiang Xu, Wei Liu, Fenggao Lang, Yuanyuan Zhang, and Chenglong Wang. Distance measurement model based on rssi in wsn. *Wireless Sensor Network*, 2(8):606–611, 2010. 73, 91, 135, 210

Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. *The 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05), Chicago, IL, USA, May 25 - 28*, pages 46–57, 2005. 184

Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. *The 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, July 16-21*, pages 70–84, 2001. 56

Yingyue Xu and Hairong Qi. Mobile agent migration modeling and design for target tracking in wireless sensor networks. *Ad Hoc Networks*, 6(1):1–16, 2008. 12, 63

Zhe Yang and Abbas Mohammed. A survey of routing protocols of wireless sensor networks. Technical report, Blekinge Institute of Technology Sweden, 2010. 43

Chien Yao, Po-Ning Chen, Tsang-Yi Wang, Yunghsiang S. Han, and Pramod K. Varshney. Performance analysis and code design for minimum hamming distance fusion in wireless sensor networks. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 53(5):1716–1734, May 2007. 132

Muneer Bani Yassein, A. Al-zou'bi, Yaser Khamayseh, and Wail Mardini. Improvement on leach protocol of wireless sensor network (vleach). *Journal of Digital Content Technology and its Applications*, 3(2):132–136, 2009. 53

Fan Ye, Alvin Chen, Songwu Lu, and Lixia Zhang. A scalable solution to minimum cost forwarding in large sensor networks. *The 10th International Conference on Computer Communications and Networks, Scottsdale, Arizona, USA, October 15-17*, pages 304–309, 2001. 49, 88, 136

Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. *The 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02), New York, USA, June 23-27*, pages 1567–1576, 2002. 132, 171

Ossama Younis and Sonia Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 3(4):366–79, 2004. 54, 131

Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *Network Magazine of Global Internetworking*, 20(3):20–25, September 2006. 130

Moustafa A. Youssef, Adel Youssef, and Mohamed F. Younis. Overlapping multihop clustering for wireless sensor networks. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, 20(12):1844–1856, December 2009. 103

Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA Computer Science Department, 2001. 57

Wei Yuan, Srikanth V. Krishnamurthy, and Satish K. Tripathi. Synchronization of multiple levels of data fusion in wireless sensor networks. *the Global Telecommunications Conference, (GLOBECOM '03. IEEE), San Francisco, USA, December 1-5*, pages 221–225, 2003. 9

S. A. R. Zaidi, M. Hafeez, S. A. Khayam, D.C. Mclernon, M. Ghogho, and K. Kim. On minimum cost coverage in wireless sensor networks. *The 43rd Annual Conference on*

Information Sciences and Systems (CISS 09), Johns Hopkins University, Baltimore, MD, March 18-20, pages 213–218, 2009. 103

Xiangbin Zhu and Wenjuan Zhang. A mobile agent-based clustering data fusion algorithm in wsn. *International Journal of Electrical and Computer Engineering*, 5(5): 227–280, 2010. 7