ENERGY-EFFICIENT ROUTING FOR WIRELESS SENSOR NETWORKS WITH A MOBILE SINK

by

Can Tunca

B.S., Computer Science and Engineering, Sabanci University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2012

ENERGY-EFFICIENT ROUTING FOR WIRELESS SENSOR NETWORKS WITH A MOBILE SINK

APPROVED BY:

Prof. Cem Ersoy ....
(Thesis Supervisor)

Prof. Günhan Dündar ....

Assoc. Prof. Tuna Tuğcu ....

DATE OF APPROVAL: 06.09.2012

ACKNOWLEDGEMENTS

Foremost, I would like to thank my thesis supervisor Prof. Cem Ersoy for his guidance in every aspect of this thesis and my academic life. He is not only a great mentor but a great friend, and I believe these two qualities are hardly found in the same person.

I thank the members of my thesis jury, Assoc. Prof. Tuna Tuğcu and Prof. Günhan Dündar, for their invaluable comments, which definitely helped making this thesis better.

I deeply thank Sinan Işık and M. Yunus Dönmez for sharing their knowledge and experience with me. They supported me both morally and practically through all the stages of my thesis. Without them this work would not be as it is.

I also would like thank my friends and colleagues in NETLAB, especially Halil, Yunus Emre, Bilgin, Orhan, Itır, for the joyful moments and discussions we had. Working in the same environment with them has certainly made it easier for me to complete this thesis.

I want to sincerely thank my mother Fatmanur and my father Erdal for always showing unquestioned faith in me and supporting everything I pursued.

Finally, I thank my one and only Ceren for all the support and love she has given.

I would like to express my gratitude to TÜBİTAK (Technical Research Council of Turkey) BİDEB for the financial support through my MSc education.

This work is supported by the European Community's Seventh Framework Programme (FP7-ENV-2009-1) under the grant agreement FP7-ENV-244088 FIRESENSE.

ABSTRACT

ENERGY-EFFICIENT ROUTING FOR WIRELESS SENSOR NETWORKS WITH A MOBILE SINK

The concentration of data traffic towards the sink in a wireless sensor network causes the nearby nodes to deplete their batteries quicker than other nodes, which leaves the sink stranded and disrupts the sensor data reporting. To mitigate this problem the usage of mobile sinks is proposed. Mobile sinks implicitly provide load-balancing and help achieving uniform energy-consumption across the network. However, advertising the location of the mobile sink to the network introduces an overhead in terms of energy consumption and packet delays. In this thesis, we propose a new routing protocol, Ring Routing, which aims to minimize this overhead while preserving the advantages of mobile sinks. It is a distributed, energy-efficient mobile sink routing protocol that minimizes data reporting delays, and hence is suitable for real-time applications. We evaluate the performance of Ring Routing via extensive simulations conducted in OPNET.

ÖZET

HAREKETLİ ÇIKIŞ DÜĞÜMLÜ KABLOSUZ ALGILAYICI AĞLAR İÇİN ENERJİ GÖZETEN YÖNLENDİRME

Kablosuz algılayıcı ağlarda, algılayıcı verilerinin çıkış düğümü etrafında yoğunlaşması bu bölgedeki düğümlerin pillerini çabuk bitirmelerine ve erken ölümlerine yol açar. Bu ölümler çıkış düğümlerinin ağla olan bağlantılarını koparır ve algılayıcı verilerinin toplanmasını engeller. Bu sorunu ortadan kaldırmak için hareketli çıkış düğümü kullanımı önerilmektedir. Hareketli çıkış düğümü, ek bir çabaya gerek duymadan ağda yük dağılımı ve tekdüze enerji sarfiyatı sağlar. Ancak hareketli çıkış düğümünün konumunu taze bir şekilde ağın geneline bildirmek, hem enerji hem de paket gecikmeleri açısından ek yüke yol açmaktadır. Bu tezde, bu ek yükü en aza indirgerken aynı zamanda hareketli çıkış düğümlerinin sağladığı avantajları koruyan yeni bir yönlendirme protokolü olan Halka Yönlendirme'yi öneriyoruz. Dağıtık ve enerji gözeten bir yönlendirme protokolü olan Halka Yönlendirme aynı zamanda veri iletim gecikmelerini en aza indirgemeyi hedefler ve bu özelliğiyle gerçek zamanlı uygulamalar için kullanılabilir. Halka Yönlendirmenin başarımını değerlendirmek amacıyla OPNET ortamında geniş kapsamlı benzetim deneyleri koşturulmuştur.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ADV	Advertisement
ALURP	Adaptive Location Updates for Mobile Sinks
AN	Anchor Node
ANHT	Anchor Node History Time
ANPI	Anchor Node Position Information
ANPIREQ	Anchor Node Position Information Request
ANPIRESP	Anchor Node Position Information Response
ANPIS	Anchor Node Position Information Share
ANS	Anchor Node Selection
CTS	Clear To Send
DAG	Directed Acyclic Graph
DDB	Dynamic Directed Backbone
DDRP	An Efficient Data-Driven Routing Protocol
DHA	Data Dissemination Based on Home Agent and Access Node
DQM	Data Quality Maximization
EA	Expect Area
EAR2	Expect Area Based Real-Time Routing Protocol
EEMSRA	Energy-Efficient Mobile Sink Routing Algorithm
EG	Expect Grid
EGRR	Real-Time Routing Protocol Based on Expect Grids
GBEER	Grid-Based Energy Efficient Routing
GHT	Geographic Hash Table
GPS	Global Positioning System
GRAB	Gradient Broadcast
HCDD	Hierarchical Cluster-Based Data Dissemination
HPDD	Hexagonal Path Data Dissemination
LBDD	Line-Based Data Dissemination
LPL	Low Power Listening

MAC	Medium Access Control
MQR	Multi-Tier Grid Routing
QDD	Quad-tree Based Data Dissemination
RC	Ring Change
RLW	Random Line Walk
RPL	A Hybrid Routing Protocol for WSNs with Mobile Sinks
PRM	Preamble
RTS	Ready To Send
SEAD	Minimum-Energy Asynchronous Dissemination
SNR	Signal-to-Noise Ratio
STD	Storage Delay Threshold
TDMA	Time Division Multiple Access
TTDD	Two-tier Data Dissemination
UAV	Unmanned Aerial Vehicle
WSN	Wireless Sensor Network

1. INTRODUCTION

Energy efficiency is the most important issue for wireless sensor networks (WSN) since sensor nodes have limited batteries. Replacing the batteries of sensor nodes is likely to require significant effort; therefore, WSNs have to be able to operate without human intervention for an adequately long time. In WSNs with static (immobile) sinks, the nodes close to the sinks are more likely to deplete their battery supplies before other nodes due to the intersection of multi-hop routes and concentration of data traffic towards the sinks. This problem is referred to as the energy hole problem [1, 2]. Node deaths would lead to disruptions in the topology and reduction of sensing coverage. Moreover, sinks could become isolated and sensor data generated across the network would no longer be obtained. Therefore, routing protocols designed for immobile sinks have to incorporate load-balancing in order to achieve uniformity of energy consumption throughout the network. The usage of mobile sinks is proposed and explored as a possible solution to this problem [3–8].

Mobile sinks implicitly provide load balancing without extra effort [9]. The hotspots around the sink change as the sink moves, and the increased energy drainage around the sink is spread through the network which helps achieving uniform energy consumption. Uniform energy consumption extends the network lifetime.

Other advantages of sink mobility include security benefits. Compromise of mobile sinks are much more difficult than immobile sinks. An adversary would have to locate and chase down a mobile sink carrier to damage the sink or retrieve any sensitive information. Moreover, mobile sinks could retrieve sensor data from isolated portions of a network which might otherwise be inaccessible in a static sink case, thus enhancing network connectivity [10, 11].

The advantages of mobile sinks do however come at a cost. Advertising the changing location of the sink freshly across the network is not trivial. The overhead of this operation should not exceed a certain limit in order not to diminish the advantages of energy savings introduced by the usage of mobile sinks. The routing protocols designed for WSNs with mobile sinks should minimize the energy overhead of frequent sink location advertisement while avoiding an extreme increase in the sensor data delivery latencies. Especially for real-time WSN applications, the validity of the sensor data depends on its freshness.

An example usage scenario for WSNs with mobile sinks is fire detection systems [12]. A fire detection system consists of numerous sensor nodes deployed on a forest area and one or several mobile sinks. Sensor nodes report temperature or humidity in a periodic manner. In case of a fire, the sensors detecting a drastic change of sensor values go into alarm mode and increase their reporting frequency. The mobile sink(s) could be placed on a motorized vehicle or be carried by a human. The sink could then be moved across the forest to gather the periodic reports generated by the sensors or around the forest if the terrain is not suitable for navigation. Mobile sinks carried by fire-fighters would assist them in their efforts towards extinguishing the forest fire by providing them with fresh and detailed information about the area of interest.

Other application scenarios of mobile sinks include habitat monitoring. The mobile sink might be deployed on a robot that collects information from the sensors deployed on different areas of a large field [13]. Battlefield surveillance, where sensors detect and monitor enemy troop or vehicle movements, is also an applicable scenario. In such a scenario static sinks are not preferred since they can easily be located and compromised by an adversary [10]. Unmanned aerial vehicles (UAV) may also be used to collect the harvested intelligence. Traffic monitoring, smart houses and hospitals, pollution control are other example scenarios [4].

The most naive approach at mobile sink routing is flooding the sink location periodically across the network. However, such an approach would introduce an immense overhead since every node in the network frequently transmits and receives broadcasts. Two-tier routing protocols aim to decrease this overhead by selecting specialized sensor nodes that acquire the sink position freshly while other nodes retrieve the sink position information from these specialized nodes whenever required (i.e. sensor data is available). This approach eliminates the need to advertise the sink's position to all nodes in the network and decreases the overall energy consumption significantly. Moreover, the need for broadcasts are minimized. Broadcasts are highly energy-inefficient since every node in the transmitter's neighborhood have to remain awake, receive the transmitted data and process it.

The two-tier architecture decreases the overall energy consumption of the network significantly. However, since the specialized second-tier nodes are likely to handle more traffic and hence spend more energy, hotspot problems arise. In order to avoid hotspots, either the second-tier nodes have to be replaced with regular nodes during the WSN operation or the number of second-tier nodes have to be relatively large to decrease the extra load on each second-tier node by distributing it over a number of nodes.

In this thesis, we propose a new two-tier hierarchical routing protocol for WSNs with a mobile sink, named Ring Routing. Some highlights and key features of Ring Routing are listed as follows:

- Ring Routing establishes a second-tier ring structure which allows the fresh sink location to be easily delivered to the ring and regular nodes to acquire the sink location from the ring with minimal overhead whenever needed.
- The ring structure can be easily changed, meaning that the second-tier ring nodes are able to switch roles with regular nodes by a straightforward and efficient mechanism, thus mitigating the hotspot problem.
- The mobile sink selects anchor nodes among sensor nodes in its vicinity along its path and these anchor nodes relay sensor data to the sink. In case the sink location information obtained by a sensor node loses its freshness, the previous anchor node relay sensor data to the current anchor node, and therefore, packet losses are prevented. This mechanism is based on the technique referred to as progressive footprint chaining [14].

- Ring Routing relies on minimal amount of broadcasts; therefore, it is applicable to be used for sensors utilizing asynchronous low-power MAC protocols designed for WSNs [15].
- Ring Routing does not have any MAC layer requirement except support for broadcasts. It can operate with any type of energy-efficient, duty cycling MAC protocol (synchronized or asynchronized).
- Ring Routing is suitable for both event-driven and periodic data reporting applications. It is not query-based so that data are disseminated reliably as they are generated.
- Ring Routing provides fast data delivery due to the quick accessibility of the proposed ring structure, which allows the protocol to be used for real-time applications.
- No information about the motion of the sink is required for Ring Routing to operate. It does not rely on predicting the sink's future position, and is applicable to scenarios where the sink's motion is random.

Ring Routing uses greedy geographic routing as the underlying routing solution.

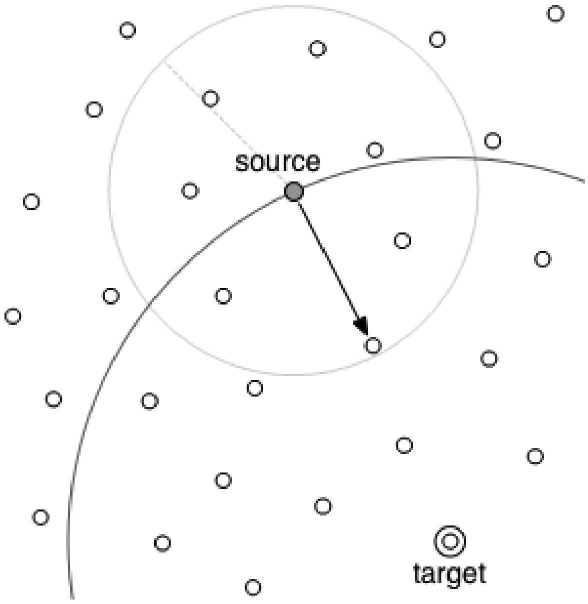


Figure 1.1. Greedy geographic routing.

Geographic routing is regarded as highly scalable and energy-efficient; therefore, it is an attractive routing solution for WSNs with position-aware sensors [16, 17]. Geographic routing only requires local knowledge to operate. Nodes, at each hop, forward data to the neighbors that are the closest to the destination position. Geographic routing is illustrated in Figure 1.1. To overcome the difficulties in finding routes in case of topology defects, for instance routing around voids, many protocols which extend geographic routing have been proposed [18–20].

With all these properties, Ring Routing is an energy-efficient, reliable routing protocol that provides fast data delivery. Extensive simulations are conducted to evaluate the performance of Ring Routing. It is compared with two existing mobile sink routing protocols, Line-Based Data Dissemination (LBDD) [21] and Railroad [22], in terms of energy consumption and reporting delay metrics. Ring Routing outperforms its competitors in almost all scenarios and proves to be a successful routing solution.

The thesis is organized as follows: In Chapter 2, the literature review on routing for WSNs with mobile sinks is provided. Details of Ring Routing are given in Chapter 3. The simulation results and performance evaluation of Ring Routing are presented in Chapter 4. The thesis is concluded in Chapter 5 with final remarks and future research directions.

2. LITERATURE REVIEW

2.1. Routing in Wireless Sensor Networks

Wireless sensor networks consist of sensors with short radio communication ranges deployed on a large area and employ multi-hop communications since communicating with the sink directly over a large distance would lead to extreme energy expenditures. Routing protocols designed for wireless sensor networks aim to establish multi-hop data dissemination routes to the sink. A naive approach would be to construct a static tree including all the sensor nodes in the network rooted at the sink. Such an approach would fail to mitigate the congestion around the sink which would create hotspots and result in early node failures due to battery depletions. Therefore, a successful WSN routing protocol employs load-balancing to establish uniform energy consumption across the network thus elongating the network lifetime.

An important approach to the WSN routing problem is to route data along multiple paths. Directed Diffusion [23] is an important example of this approach. The sink issues interests (data requests) across the network which allow the sensor nodes to record the direction from which the interests come as gradients. Since the interests are flooded, there are likely to come from different neighboring nodes and thus multiple gradients are recorded around a node. Data reports then flow from the sources to the sink along multiple gradient paths. Using multiple paths for data dissemination increases reliability and distributes the load on multiple nodes, but redundancy of data packets may cause an inefficiency for the total energy consumption of the network.

Another approach is to establish the routing tree intelligently by taking the loads on the nodes into account. In [24], the authors propose a load-balancing scheme which cumulatively calculates the loads of the branches of a routing tree. At each step the algorithm selects the branch with the lightest load, and then connects the node with the heaviest load to this branch. Hence the load-balanced tree is constructed iteratively.

In [25], a load-balancing scheme based on geographic routing is proposed. Each node periodically beacons its traffic levels and its distance to the sink to its neighbors. The data disseminating nodes select the next forwarding hop using this information. The scheme operates locally without the establishment of a routing tree. An Adaptive Load-Balanced Algorithm for Geographic Forwarding [26] proposes a similar load-balancing scheme which employs the sharing of the traffic levels and the distance to the sink information in the MAC layer. Instead of periodic beaconing, this information is acquired during the data transmission process encoded in the MAC layer messages.

In [27], the problem of determining load-balanced routes is defined as a network optimization problem. The objective function tries to minimize the overall energy cost of the constructed data paths to the network. Moreover, a distributed algorithm is proposed for finding optimal data flows, which is analogous to satisfying Kircchoff's Voltage and Current Laws in an electric circuit.

Detailed reviews of various other approaches and numerous important existing WSN routing protocols are included in [16, 17].

2.2. Mobile Sink Routing Protocols

There are many approaches to the problem of routing in WSNs with mobile sinks; the most important and the most widely adopted one being the hierarchical two-tier approaches. Hierarchical two-tier approaches aim to decrease the load of advertising the sink's position to the network by establishing a second-tier structure of nodes which acquire the sink position information foremost. Other nodes query the second-tier nodes to acquire the sink position information whenever necessary. The routing protocol proposed in this thesis, Ring Routing, is also a hierarchical two-tier routing protocol.

In this section, the most prominent two-tier hierarchical and non-hierarchical mobile sink routing protocols are reviewed and their benefits and drawbacks are determined. Lastly the highlights of Ring Routing with respect to these protocols are put forward.

2.2.1. Review of Hierarchical Two-tier Routing Protocols

Two-Tier Data Dissemination in Large Scale WSNs (TTDD) [28] is one of the predecessors of the two-tier approach. It is a hierarchical virtual grid based approach which is source-oriented in the sense that each source node with valid sensor data proactively constructs a grid around itself and becomes a crossing point of this grid. The grid covers the whole network (Figure 2.1a). For grid construction to be possible, position-awareness of sensor nodes is required. Whenever sinks require data, they query the network by local flooding within a cell (defined by the grid constructed by a source) and these queries are relayed to the source node. Data is then forwarded to the sink using the reverse of the path taken by the data request. Progressive footprint chaining strategy is used to make the mobility of the sink within a cell transparent to the network. For periodic data reporting applications where every sensor in the network report data, the overhead of constructing grids (one grid for each node) is immense. In order to lighten this burden, nodes may select aggregation nodes among themselves to decrease the number of source nodes constructing grids.

Grid-Based Energy-Efficient Routing From Multiple Sources to Multiple Mobile Sinks (GBEER) [29] is a hierarchical virtual grid-based method similar to TTDD. However, unlike TTDD, it constructs a single combined grid structure for all possible sources. To build the grid structure location-awareness of sensor nodes is necessary. Both data requests originated from the sink and data announcements originated from the source are propagated through the grid structure. The concept of Quorums is proposed to ensure that these different types of packets intersect at a header (grid-point). Data announcements are propagated horizontally along the grid while data requests are propagated vertically, ensuring that these packets intersect at a crossing point. The position of the sink is then delivered to the source node, and data is delivered directly to the sink. Like TTDD, progressive footprint chaining is used to render the mobility of the sink transparent at the grid.

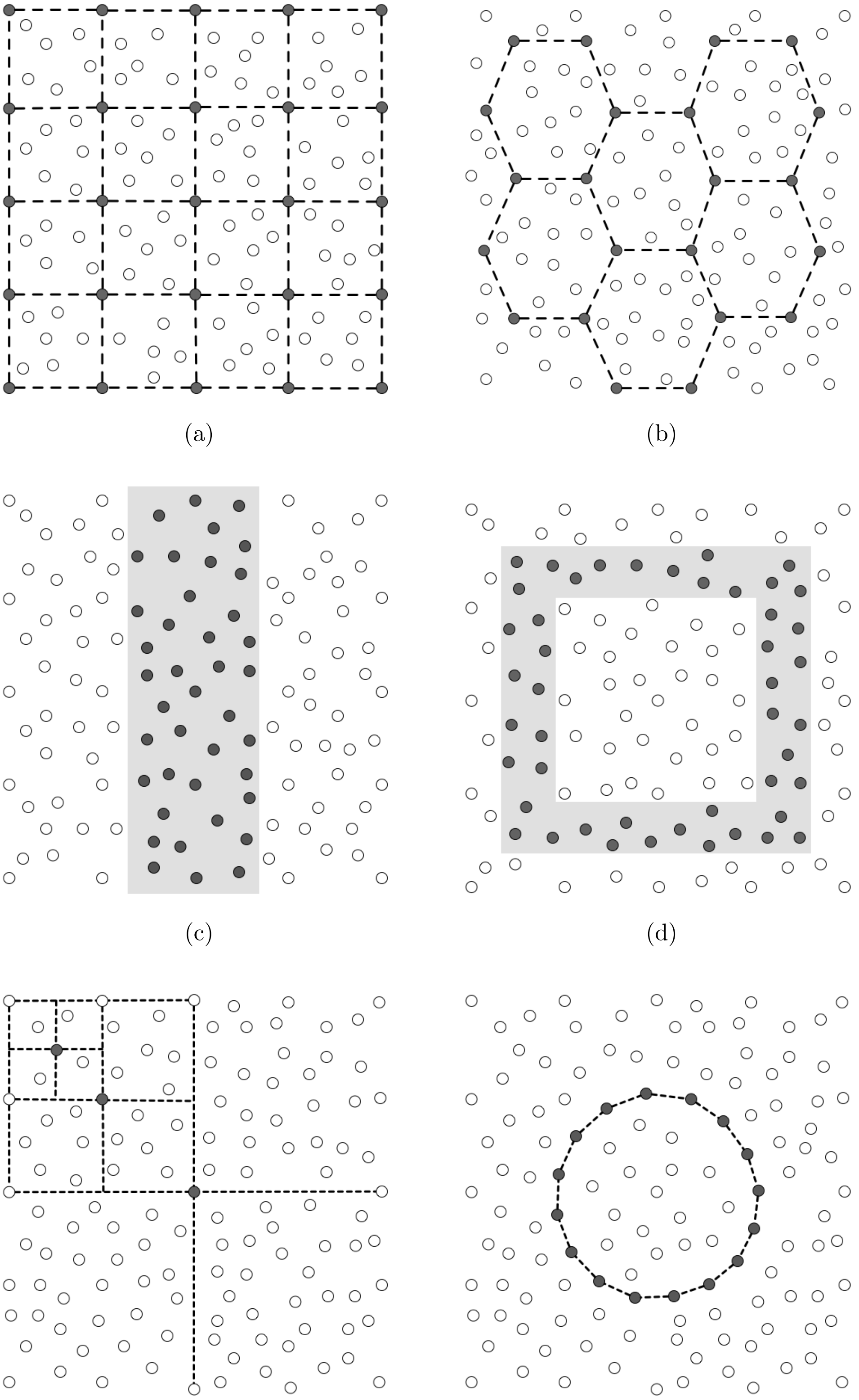
In place of a rectangular grid used in TTDD and GBEER, other similar structures might be utilized. Hexagonal Path Data Dissemination (HPDD) [30] uses a common grid composed of hexagons (Figure 2.1b) which is shown to outperform rectangular grid based approaches.

Hierarchial Cluster-Based Data Dissemination (HCDD) [31] is a hierarchical approach which uses clustering to determine second-tier nodes. Like GBEER and HPDD a combined hierarchical structure for all data sources is constructed. The cluster heads are called Routing Agents which are responsible for propagation of data requests. Max-Min D-Cluster Formation Algorithm is used for determining cluster heads. The advantage of this algorithm is its ability to operate without position-information of sensor nodes. Other aspects of this approach resemble TTDD.

Minimum-Energy Asynchronous Dissemination to Mobile Sinks in WSNs (SEAD) [32] uses a minimum-cost weighted Steiner tree as the second-tier structure which selects replicas at intermediate points. SEAD also uses progressive footprint chaining to make the sink mobility transparent to the overlaying structure. Position-awareness of the sensor nodes is required for construction of the Steiner tree and data dissemination.

Quad-tree Based Data Dissemination Protocol (QDD) [33] partitions the network into successive quadrants (Figure 2.1e). The center point of each quadrant becomes a second-tier node. The quadrants are recursively divided further into smaller quadrants until the resolution of the second-tier nodes are sufficient for quick access to the virtual structure. Data announcements and queries are sent to the center points of quadrants in a recursive manner until they rendezvous.

Geographic Hash Table (GHT) [34] hashes the data type into geographic coordinates. Nodes deliver their data to the nodes closest to the resulting geographic positions, and sinks may retrieve data by sending a query towards the geographic position determined by using the same hash function with the requested data type.



(a) (b)

(c) (d)

(e) (f)

Figure 2.1. Various two-tier structures: (a) TTDD/GBEER (b) HPDD (c) LBDD (d) Railroad (e) QDD (f) Ring Routing.

Dynamic Directed Backbone (DDB) [35] and Efficient Routing proposed by Fodor and Vidacs [36] construct a backbone as the second-tier structure. The backbone is composed of leader and gateway nodes. Leader nodes form clusters of nodes in their own neighborhood and coordinate data traffic associated with all nodes in their clusters. Leader nodes communicate with each other by gateway nodes which complete the connectivity of the backbone structure. The sink connects to the backbone which extends through the network, and data dissemination is performed over the backbone.

Data Quality Maximization (DQM) [37] is another routing protocol based on a backbone consisting of gateways. This protocol assumes predictability of the sink movement and selects gateways adjacent to the predicted path of the sink. The sensors establish shortest path routes with the gateways using Floyd-Warshall algorithm. Gateways aggregate incoming data and wait for the sink to arrive.

Multi-tier Grid Routing (MQR) [38] combined the grid based approach with clustering. A recursive grid is constructed similar to the quad-tree approach of QDD. However, the crossing points are not data dissemination centers. Within each grid cell, a distributed clustering algorithm is executed and the clusterheads are selected as data aggregator nodes. Grid cells have binary addresses, hence the recursive structure of the multi-tier is clearly represented and cells are easily accessible by the sink.

Energy-Efficient Mobile Sink Routing Algorithm (EEMSRA) [39] is another clustered routing protocol based on LEACH [40]. LEACH proposes a method to randomly select cluster-heads. These clusterheads are used as gateways to the sink. The clusterheads change periodically in order to mitigate the hotspot problem. A key feature of EEMSRA is that the clusterheads create a TDMA (time division multiple access) schedule informing each node in the cluster of when they can transmit data. In this sense, EEMSRA is a cross-layer protocol which operates coordinated with the MAC layer. The clusterheads also perform aggregation before transmitting data to the sink. The sink broadcasts its next projected cluster visit in order to enable the network to update routes prior to the sink's actual arrival at the cluster. This approach, while saving significant amounts of energy, requires the sink to at least have knowledge about its short-term trajectory. Moreover, a sink movement scheme which takes the clusters' energies into account has been proposed. The sink always selects the neighboring cluster with the maximum remaining energy as the next visiting location. Considering these approaches, EEMSRA employs controlled sink mobility.

Real-Time Routing Protocol Based on Expect Grids (EGRR) [41] is a grid-based routing protocol which extends Expect Area Based Real-Time Routing Protocol (EAR2) [42]. In EAR2 the source nodes calculate Expect Areas (EA) using the current speed and position of the sink and send data to the closest node in the EA. The closest node then floods data within the EA, and data is disseminated to the sink. The authors of [41] aim to minimize the flooding overhead of EAR2 by introducing Expect Grids (EG) in addition to EAs. The closest node in an EA receiving data forwards it through the EG constructed in the EA. The sink retrieves data from the closest crossing point of the grid. By this mechanism extensive broadcasts are avoided.

Line-based Data Dissemination (LBDD) [21] defines a vertical strip of nodes which divide the field of deployment into two equal portions (Figure 2.1c). The nodes on this strip are referred to as in-line nodes. Sensor data are sent to the line and the first in-line node encountered stores the data. The sink sends a data query to the line and the query is propagated through the line until the in-line node storing the data is reached. The in-line node then forwards the data directly to the sink, and data dissemination is completed.

Railroad [22] constructs a virtual infrastructure called the rail. The rail is a closed loop of a strip of nodes which has the shape defined by the outline of the network (Figure 2.1d). The nodes on this rail are called rail nodes. When a node has sensor data, it sends information about this data (meta-data) to the nearest rail node. The rail node receiving the meta-data constructs a station which is a portion of the rail centered on the rail node with minimum width of communication range. The meta-data is shared among the nodes residing on the station. The sink queries the rail for meta-data and

Table 2.1. Summary and comparison of two-tier routing protocols.

Protocols	**Position-awareness**	**Sink mobility**	**Second-tier structure**	**Hotspot mitigation (strength)**
TTDD [28]	Yes	Random	Rect. grid	Multiple grids for each source (strong)
GBEER [29]	Yes	Random	Rect. grid	Structure change (average)
HPDD [30]	Yes	Random	Hexagonal grid	Structure change (average)
HCDD [31]	No	Random	Min D-Clusters	Structure change (weak)
SEAD [32]	Yes	Random	Steiner tree	Structure change (weak)
QDD [33]	Yes	Random	Quad-tree	N/A
GHT [34]	Yes	Random	Hashed geo-coordinates	N/A
DDB [35]	Yes	Random	Backbone	Structure change (strong)
Fodor [36]	Yes	Random	Backbone	Structure change (strong)
DQM [37]	Yes	Predictable	Backbone	N/A
MQR [38]	Yes	Random	Grid & clusters	N/A
EEMSRA [39]	Yes	Controlled	TDMA clusters	Structure change (strong)
EGRR [41]	Yes	Predictable	Expect areas & grids	Structure change (strong)
LBDD [21]	Yes	Random	Line (wide)	Large structure (strong)
Railroad [22]	Yes	Random	Rail (wide)	Large structure (strong)
Ring Routing (proposed)	Yes	Random	Ring (one-node width)	Structure change (strong)

when a station node is reached it informs the source of the sink's position and then the source node can send the corresponding data directly to the sink. One key difference of Railroad from LBDD is that queries issued by the sink travel on the rail by unicasts rather than broadcasts. To ensure that a query encounters a node with meta-data, stations must cover the width of the rail.

The above-mentioned two-tier mobile sink routing protocols are summarized and compared in Table 2.1. The second-tier structures and the hotspot mitigation strategies along with their strengths are also provided.

2.2.2. Review of Non-hierarchical Routing Protocols

Column-Row Location Service [43] uses a similar approach to GBEER in the sense that queries by the sink are propagated vertically while data announcements are propagated horizontally across the network. However, it does not utilize a grid structure, so that there is no virtual infrastructure and all nodes contribute in distribution of routing information packets. In order for packets to travel along vertical and horizontal paths, position-awareness of sensor nodes is required.

Double-Blind Data Discovery Using Double Cross [44] is a mobile sink routing protocol based on Random Line Walk (RLW) [45]. In RLW, when a sensor node has data, it forwards it towards a random direction which is maintained in the successive hops, thus forming a line. In order for this mechanism to function, the relative direction information of the neighboring nodes is sufficient. This information can be derived by the simple cosine rule in case the distances of the neighbors are known. Therefore, RLW does not require position-awareness of sensor nodes, and distances may be inferred by observing SNR values. RLW ensures that a data packet propagates along a straight line; however, it cannot guarantee for the paths of two separately generated data packets to have a predetermined angle between them, which would require position-awareness of sensor nodes (e.g. vertically and horizontally propagated packets in Column-Row Location Service). Double Cross extends the idea of RLW by exploiting the simple geometric property of a planar: The intersection probability of two pairs of orthogonal lines in a planar is more than 99%. When a node has data, it sends it along two orthogonal lines which correspond to four directions. The queries issued by the sink are sent in four directions in the same manner. Due to the geometric property given above, data and queries are bound to meet, thus data dissemination is completed.

An Efficient Data-Driven Routing Protocol (DDRP) [46] exploits overhearing of data packets for the sensor nodes to gratuitously learn the routes toward the sink. Each data packet carries an additional information on the hop distance of the sender node to the sink. The overhearing of such a packet will provide other nodes in the communication range with a route to the sink. As data packets are generated and transmitted, the knowledge of routes to the sink propagates across the network. If a sensor node has not yet learned of a valid route to the sink, it uses random walk mechanism to forward its data towards a random direction until the packet encounters a node with a route to the sink or the sink itself. To reduce the frequency of random walk procedure, which is inefficient, DDRP enforces data packets to be buffered for a certain amount of time before they are transmitted, in order to take the chance that a route is learned in the meantime.

In Gradient Broadcast (GRAB) [47], the sink builds a cost field by flooding advertisement (ADV) packets across the network. The cost of a node is defined as the estimated energy overhead to forward a packet from this node to the sink. The authors assume that a node can estimate the energy cost of sending data to its neighbors by observing the SNR values of the neighbors' transmissions. Nodes include their own costs in the ADV packets, and hence nodes can update their costs by accumulating the cost encoded in the ADV packet and the local cost for communicating with the transmitting neighbor. The cost field implicitly designate the global direction towards the ring. When a source node has data, it does not select a single recipient. Rather, data packets are broadcast and the nodes with costs lower than the transmitter continue propagation. Data packets are funneled down to the bottom of the cost field where the sink resides. The sink observes the success ratios of the received packets and the average consumed energies and compares them with the past values in order to determine if the cost field needs to be rebuilt, which involves broadcasting a new ADV packet to be flooded across the network.

Data Dissemination Protocol Based on Home Agent and Access Node (DHA) [48] extends and generalizes the progressive footprint chaining mechanism to handle both sink position advertisement and data dissemination. Two specialized nodes, referred to as Home Agent and Access Node, are employed. The home agent represents the mobile sink to the sensor nodes, thus making the movement of the sink transparent to the network. On the other hand, the access node represents the mobile sink to the home agent. Only the home agent and the access node are affected by the movement of the sink. As the sink moves, it selects new access nodes which inform the home agent of their new roles. The home agent aggregates and relays data packets to the access node which relays them directly to the sink. Therefore, source nodes only need to know the position of the home agent to disseminate data. The home agent is changed occasionally to avoid hotspots, and the position of the new home agent is flooded across the network.

TwinRoute [49] provides a hybridization of two basic routing approaches: Proactive Scheme, Reactive Scheme. Proactive scheme exploits the predictability of sink mobility. It keeps record of sink visits at each sensor node to choose storage-nodes. The storage delay threshold (STD) parameter is used to determine the storage-nodes. Storage-node rooted dissemination trees are constructed to relay data to these nodes rather than the sink itself. As a mobile sink passes a storage-node, it collects the aggregated data. Reactive Scheme is similar to flooding, but it is limited by the parameter tree depth. Also nodes having participated in a tree construction previously do not participate in upcoming attempts within a tunable timer. TwinRoute forces a hybridizing condition by introducing an additional parameter called scheme preference parameter which denotes the distance gain necessary to switch from P-scheme to R-scheme. Storage nodes exist, however nodes could choose to send their packets directly to the sink whenever appropriate. TwinRoute is suitable for delay-tolerant applications. Position-awareness of sensor nodes is not required.

Adaptive Location Updates for Mobile Sinks (ALURP) [50] is a non-hierarchical method which does not require position-awareness of sensor nodes. When the network is deployed, the mobile sink advertises itself to the entire network by flooding. Nodes forward data packets to a local area (defined by a specified radius around the sink) rather than the sink itself, while nodes within this local area route data packets directly to the sink. As the sink moves, it re-advertises itself by flooding only inside this local area (which is a few hops wide). When the sink moves out of this area, it needs to advertise itself to the whole network again by flooding. By employing a local area around the sink, ALURP decreases the frequency of global flooding at the expense of increased delay caused by the usage of suboptimal routes to the sink.

A Hybrid Routing Protocol for WSNs with Mobile Sinks (RPL) [51] is similar to ALURP in the sense that the sink proactively advertises itself to the network by global flooding and the frequency of global flooding is reduced by defining a few hops wide area in which broadcasts are confined. However, RPL defines the local area around the source nodes rather than around the sink. The global flooding of sink advertisement packets creates a directed acyclic graph (DAG). Members of an active DAG can disseminate data to the sink. A source node which is not a member of an active DAG issues a route request which is propagated in the defined local area to reach a DAG member node. If such a node is not found after a maximum number of retries, the request is flooded to the entire network. The intermediate nodes forwarding the reply to the source node record the sink position information and become DAG members for future data dissemination.

2.2.3. Benefits and Drawbacks of Existing Protocols

Grid-based protocols are advantageous for the easy-accessibility of the grid structure. Both the source nodes and the sinks can reach the grid with minimal number of hops. However, construction of the grid is non-trivial. TTDD suffers from the high overhead of constructing a grid for each source node especially in applications where many sensor nodes generate data. GBEER aims to eliminate the high overhead of constructing separate grids by establishing and maintaining a common grid structure, but the nodes making up the grid are likely to be hotspots and die quicker than other nodes. To overcome this problem the grid have to be changed from time to time which is cumbersome. Even changing a single crossing point requires informing the four neighboring crossing points which will introduce extra traffic on numerous nodes residing between the crossing points. HPDD suffers from the same problem, even though a hexagonal grid structure is better than a rectangular grid.

SEAD defines and establishes a more intelligent second-tier structure which is a Steiner tree. Even though the accessibility of this structure is better, it also suffers from the hotspot problem. Changing this structure is even more complicated and inefficient than changing a grid structure.

HCDD's advantage is that it employs a distributed clustering algorithm which can operate without position-awareness. Clustering allows a better choice of second-tier nodes; however, the distributed algorithm's overhead is high and running it again in case the batteries of clusterhead nodes are about to deplete is very inefficient.

GHT has no overhead to select the second-tier nodes since the second tier nodes are defined by the used geographic hash function. However, the number of second-tier nodes is likely to be small, thus causing concentration of data traffic on a few nodes. Changing these nodes is practically impossible unless the hash function is changed.

The overhead of constructing the quad-tree structure in QDD is also minimal; however, no countermeasure against the hotspot problem is proposed.

DDB and Efficient Routing proposed by Fodor and Vidacs propose similar backbone structures. Changing the backbone structure has relatively low overhead since only the immediate neighbors have to be informed if a backbone node switches roles with a regular node. However, in order to cover the whole network, a large backbone with many branches have to be established, which will cause redundancy of routing control packets (sink data queries and data announcements) and thus increase the overall energy consumption in the network.

DQM is only applicable in delay-tolerant applications since the selected gateways disseminate the aggregated data only when the sink is nearby. Moreover, the gateways might never be visited by the sink; therefore, a predictable sink movement scheme has to be employed which enables gateways to be selected along the projected path of the sink. Another drawback of DQM is the difficulty of mitigating the hotspot problem. Changing gateways would introduce a large overhead since the aggregated data must also be transferred to the newly selected gateway.

MQR constructs a grid structure combined with clusters which is easily accessible. However, the overhead of constructing such a complex structure is high. Also, no countermeasure against the hotspot problem is proposed.

EEMSRA forms clusters with enforced TDMA schedules to increase energy-efficiency. This approach has MAC layer requirements, so it might not be applicable to a wide range of devices. Another limitation of EEMSRA is the need for controlled sink mobility. Other than these drawbacks, EEMSRA is a beneficial protocol in terms of energy efficiency. The authors proposed a mechanism to change the clusterheads in order to mitigate the hotspot problem.

EGRR depends on predicting the sink's movement. The sink advertisement broadcasts are minimized by using expect areas combined with expect grids. The hotspot problem is mitigated by the constant change of second-tier structure; however, the overhead of the frequent re-establishment of expect grids is high.

Another method of avoiding hotspots is to define a second-tier structure with sufficiently many number of nodes. LBDD and Railroad are examples of this approach. Even though these two protocols propose easily accessible structures which aim to mitigate the hotspot problem by distributing the load on the second tier structure to numerous nodes, sharing queries or sink position advertisement packets on these structures require broadcasts and hence the total energy consumption in the network is bound to increase.

The non-hierarchical protocols eliminate the overhead of constructing a second-tier structure; however, they mostly rely on redundant packets and extensive broadcasts to advertise the sink or disseminate data.

Column-Row Location Service and Double Cross has no overhead of constructing the second-tier structure since there is no defined virtual structure. However many nodes have to be utilized along the source and the sink oriented lines across the network for each data packet in order to ensure that the sink's queries and the source's announcements meet. Packets are redundantly sent towards multiple directions. Compared to Column-Row Location Service, an advantage of Double Cross is its ability to operate without position-awareness of sensor nodes.

DDRP exploits the broadcast nature of wireless communications to acquire routes to the sink by overhearing other nodes' transmissions. Even though DDRP is straightforward and very simple to implement, overhearing increases the energy consumption significantly. Moreover, since overhearing requires all nodes in a neighborhood to listen to the medium synchronously, DDRP cannot be used with an asynchronous duty-cycling low-power MAC protocol.

DHA selects a home agent as a data aggregation and dissemination point. The load on the home agent is immense, and changing the home agent requires global flooding which is highly inefficient in terms of energy.

TwinRoute, ALURP and RPL rely on global flooding as the sink advertisement method. Even though they propose mechanisms to decrease the frequency of global sink advertisements, broadcasts across the network pose a significant inefficiency in terms of energy regardless of how rare they are used. The advantage of these protocols is that they do not require position-aware sensor nodes.

2.2.4. Ring Routing Compared to Existing Protocols

Ring Routing establishes a ring structure (Figure 2.1f) which aims to combine the easy-accessibility of grid structures (TTDD, SEAD, HPDD) with easy-changeability of backbone structures (DDB, Fodor). It also aims to reduce redundancy of routing control packets by incorporating minimal number of nodes in the ring structure, and devising a straightforward and efficient mechanism for sharing sink position advertisement packets among the ring nodes.

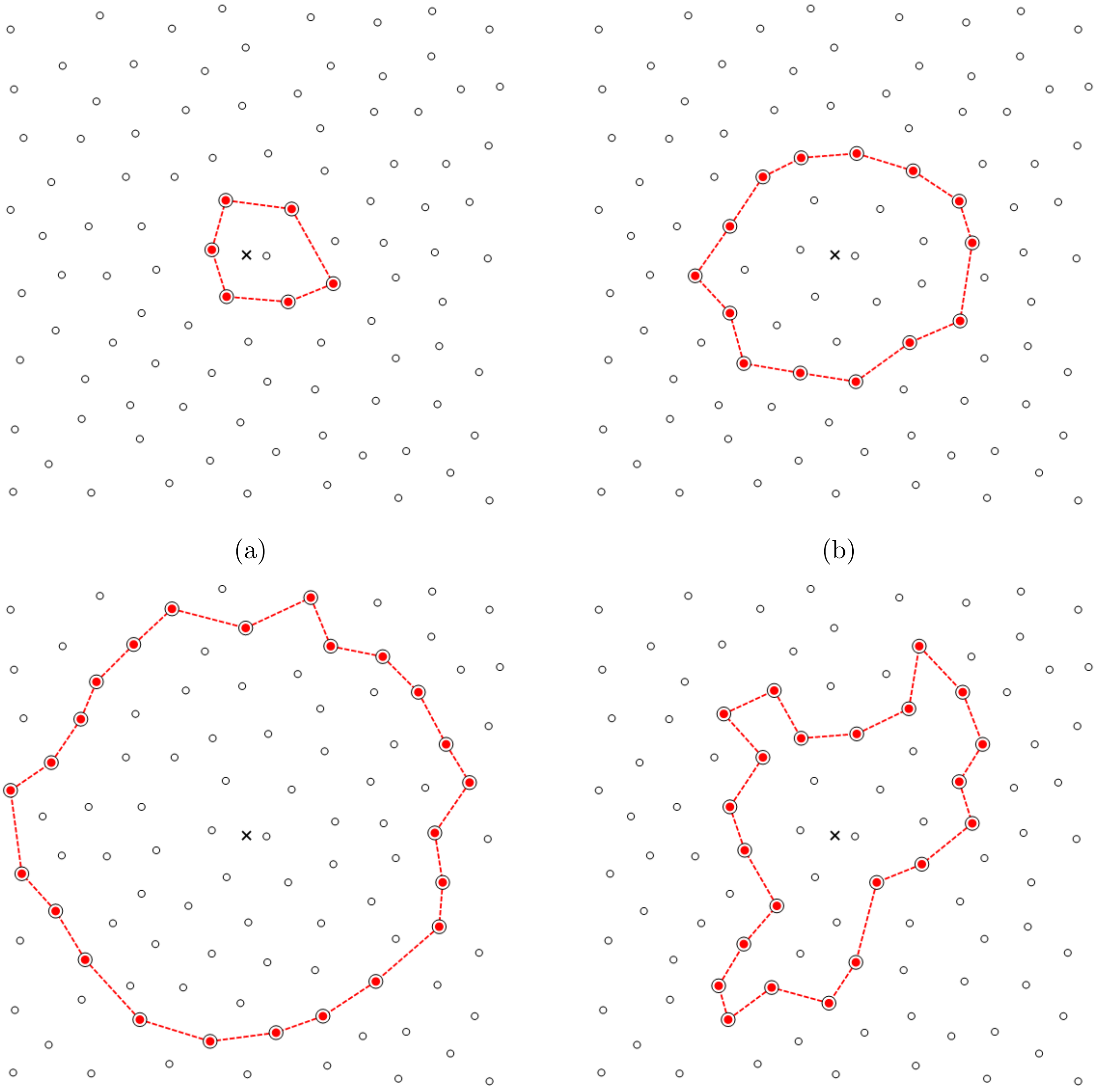
Unlike EEMSRA, Ring Routing has no MAC layer requirements and is able to operate with an asynchronous low-power MAC protocol. Moreover, it outperforms non-hierarchical mobile sink routing protocols since it relies on minimum amount of broadcast packets and employs no global flooding.

LBDD and Railroad are the most efficient among the reviewed protocols since most protocols suffer greatly from the hotspot problem. LBDD and Railroad also have minimal second-tier structure establishment overhead. Considering their advantages, these two protocols appear as strong candidates for performance comparison with Ring Routing.

3. RING ROUTING

In this chapter, we propose a new two-tier routing protocol, Ring Routing, for wireless sensor networks with a mobile sink. The two-tier structure imposes two roles on sensor nodes: regular node, ring node. Ring nodes form a ring structure which is a closed loop of single-node-width (Figure 3.1). The basis of Ring Routing is (i) advertisement of sink location to the ring, and (ii) regular nodes obtaining the sink location information from the ring whenever necessary. The two sensor roles are not static, meaning that sensor nodes can change role during the operation of the WSN. Three simple assumptions are made before going into the details of the protocol:

- Every sensor node is aware of its own position. The problem of determining the position of the nodes, which is beyond the scope of this thesis, is referred to as localization. The position information may be based on a global or a virtual local coordinate system. Localization might be achieved using a satellite based positioning system such as global positioning system (GPS) or one of the energy-efficient localization methods proposed specifically for WSNs [52–54].
- Every sensor node should be aware of the position of its neighbors. This information enables greedy geographic routing and can be obtained by a simple neighbor discovery protocol.
- The coordinates of a network center point has to be commonly known by all sensor nodes. The network center does not have to be exact and can be loaded into the sensors' memories before deployment. The ring structure encapsulates the network center at all times, which allows access to the ring by regular nodes and the sink. The network center is marked with a "X" in various example ring structures shown in Figure 3.1.



(a)

(b)

(c)

(d)

Figure 3.1. Various example ring structures: (a) small (b) medium (c) large (d) imperfect.

3.1. Ring Construction

The ring consists of a one-node-width, closed strip of nodes which are called ring nodes (Figure 3.1). As long as the ring encapsulates the pre-determined network center, it can change. The shape of the ring might be imperfect as long as the strip forms a closed loop (Figure 3.1d).

After the deployment of the WSN, the ring is initially constructed by the following simple method: An initial ring radius is determined. The nodes closer to the ring, which is defined by this radius and the network center, by a certain threshold are determined to be ring node candidates. Starting from a certain node (e.g. the node closest to the leftmost edge of the ring) by greedy geographic forwarding in a certain direction (clockwise/counterclockwise), the ring nodes are selected in a greedy manner until the starting node is reached and the closed loop is complete. The pseudo-code for this

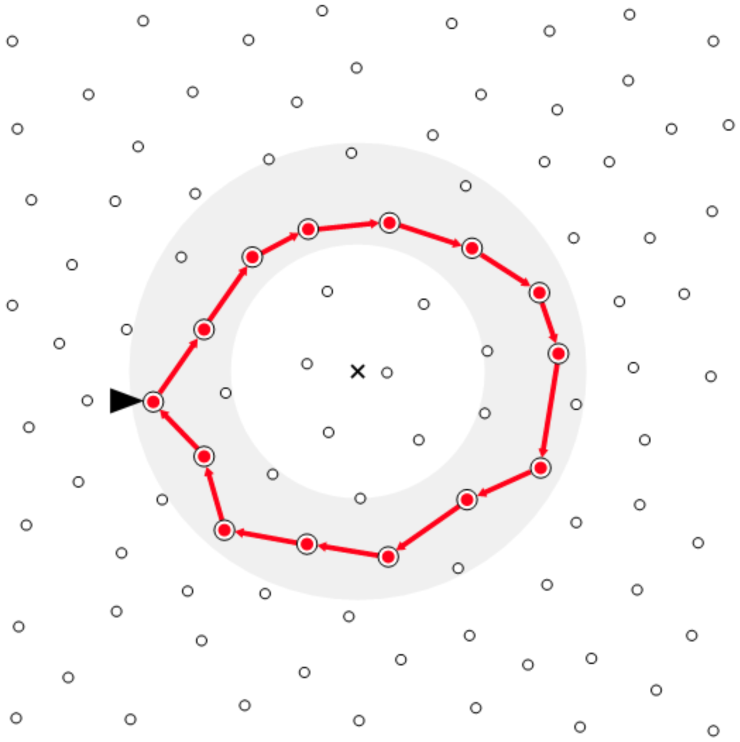


Figure 3.2. Ring construction.

```
procedure ConstructRing($startnode, ringradius, ringsearchwidth$)
    $currentnode \leftarrow startnode$
    specify $searcharea$ defined by $ringradius$ and $ringsearchwidth$
    assert that $network\ center$ is encapsulated by the ring strip of $searcharea$
    repeat
        calculate $cwregion$ within $searcharea$
        if $startnode \in currentnode.neighbors \wedge startnode$ in $cwregion$ then
            $currentnode.cwneighbor \leftarrow startnode$
            $startnode.ccwneighbor \leftarrow currentnode$
            $ringiscomplete \leftarrow true$
        else
            $nextnode \leftarrow$ the farthest neighbor in $cwregion$
            $currentnode.cwneighbor \leftarrow nextnode$
            $nextnode.ccwneighbor \leftarrow currentnode$
            $currentnode \leftarrow nextnode$
        end if
    until $ringiscomplete$
end procedure
```

Figure 3.3. Pseudo-code for the ring construction mechanism.

procedure is provided in Figure 3.3. If the starting node cannot be reached so that the loop is not completed, the procedure is repeated with selection of different greedy geographic neighbors at each hop. If after a certain number of trials the ring cannot be formed, the radius is reset to a different value and the procedure above is repeated. The initial ring construction is depicted in Figure 3.2. The starting node is marked with a black triangle. The greedy search (in the area specified by a gray shade) continues in clockwise direction until the starting node is reached again.

The initial ring construction procedure is straightforward and cheap in terms of energy. It does not require a centralized decision entity, and therefore it is applicable in a pure WSN architecture (all nodes in the network are regular sensor nodes).

After the construction of the ring, simple neighbor discovery is performed to mark the neighboring ring nodes of each sensor node. This step is crucial in order for the regular nodes to be able to access the ring. Moreover, each node should determine its position with respect to the ring (namely inside or outside the ring).

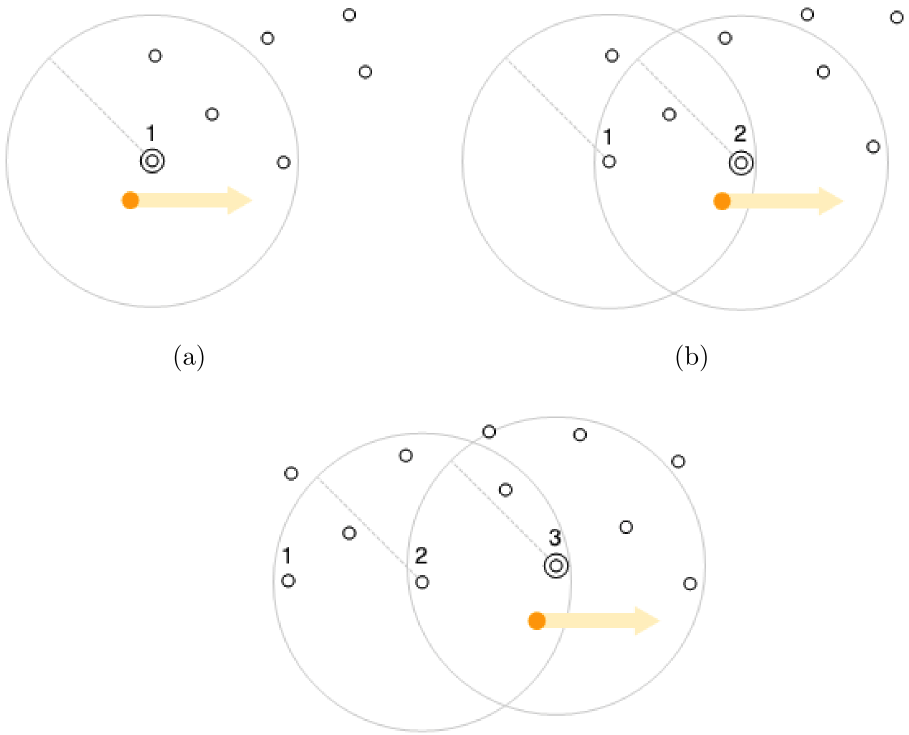
3.2. Advertisement of Sink Position

As the mobile sink moves, it selects anchor nodes among its neighbors. The anchor node serves as a delegate for managing the communications between the sink and the sensor nodes. Initially the sink selects the closest node (the node with the greatest signal to noise (SNR) value) as its anchor node, and broadcasts a anchor node selection (ANS) packet. Before the sink leaves the communication range of the anchor node, it selects a new anchor node and informs the old anchor node of the position and the MAC address of the new anchor node by another ANS packet. Since now the old anchor node knows about the new anchor node, it can relay any data which is destined for it to the new anchor node. The current anchor node relays data packets directly to the sink. This mechanism is referred to as the follow-up mechanism.

The anchor node selection mechanism is shown in Figure 3.4. First the sink selects node 1 as the anchor node. Before it leaves the communication range of node 1 it selects node 2 as the new anchor node and informs node 1. Likewise, when sink is about to leave the communication range of node 2 it selects node 3 as its current anchor node.

Thanks to the anchor node, if a source node acquires the fixed position of the current anchor node, it can directly send its data to this anchor node by geographic forwarding, and the anchor node relays this data to the sink, completing data dissemination. If the data reaches an old anchor node, the follow-up mechanism is used to relay the data to the sink.

The essence of sink position advertisement is delivering the newly selected anchor



(a)

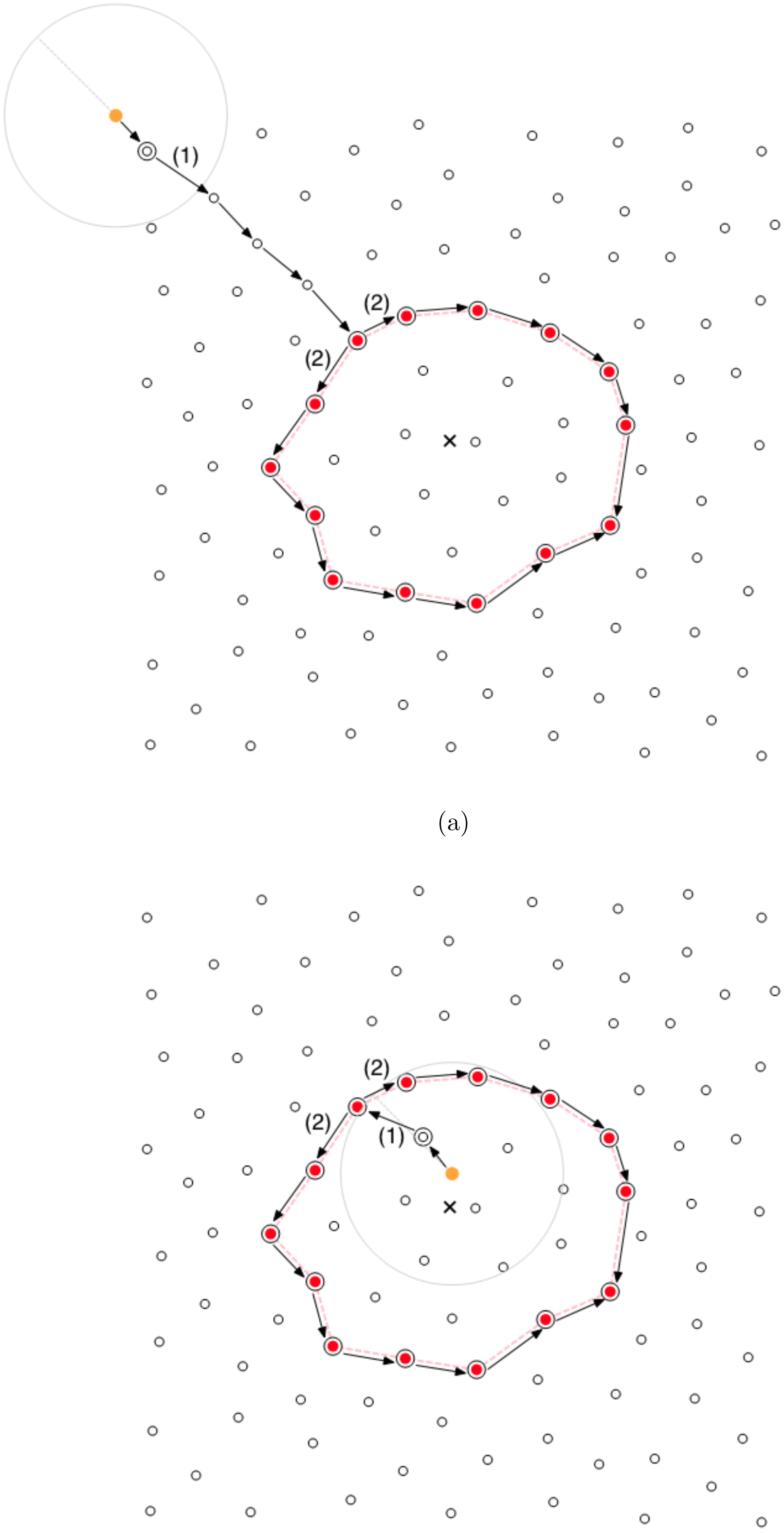
(b)

(c)

Figure 3.4. Anchor node selection: (a) time instance 1, AN: node 1 (b) time instance 2, new AN: node 2 (c) time instance 3, new AN: node3.

node's position and MAC address information to the ring. Upon selection of an anchor node, the newly selected anchor node sends an anchor node position information (ANPI) packet towards the ring. If the anchor node is outside the ring, it sends the ANPI packet towards the network center. If it is inside the ring, it sends the ANPI packet towards a point which resides on the opposite direction of the network center using greedy geographic forwarding. If the ANPI packet on its way arrives at a node which has a ring node neighbor, it is directly relayed to that ring node. The closed loop property of the ring ensures that such a ring node will certainly be reached.

After a ring node receives an ANPI packet, it shares this information by sending an anchor node position information share (ANPIS) packet to its clockwise and counter-clockwise ring neighbors. Each ring node receiving an ANPIS packet relays it to the neighbor ring node in the respective direction until the two ANPIS packets sent in the



(a)

(b)

Figure 3.5. Anchor node position advertisement. (a) sink is outside the ring (b) sink is inside the ring.

```
procedure ProcessANPIpacket($ANPIpacket$)
    if $role = regularnode$ then
        record anchor node position and MAC address information
        if $ringnode \in neighbors$ then
            send ANPIpacket to $ringnode$
        else
            $destinationPosition \leftarrow ANPIpacket.destinationPosition$
            GeographicRouting($ANPIpacket, destinationPosition$)
        end if
    else if $role = ringnode$ then
        record anchor node position and MAC address information
        create $ANPISpacket$
        set anchor node position and MAC address information in $ANPISpacket$
        send a copy of $ANPISpacket$ to $cwringneighbor$
        send another copy of $ANPISpacket$ to $ccwringneighbor$
    end if
end procedure
procedure GeographicRouting($packet, destinationPosition$)
    for all $n \in neighbors$ do
        if $distance(n.position, destinationPosition) < minDistance$ then
            $minDistance = distance(n.position, destinationPosition)$
            $targetnode \leftarrow n$
        end if
    end for
    send $packet$ to $targetnode$
end procedure
```

Figure 3.6. Pseudo-code for processing ANPI packets.

clockwise and counter-clockwise directions arrive at the same ring node. At this point all the ring nodes are aware of the position and the MAC address of the anchor node, and thus the advertisement of sink position to the ring is completed.

The sink position advertisement mechanism is depicted in Figure 3.5. The ANPI packet is sent to the ring (1), and ANPIS packet is sent around the ring (2) to share the anchor node position information with all ring nodes. The pseudo-code of the procedure executed by nodes when processing ANPI packets is provided in Figure 3.6. The pseudo-code of simple geographic routing is also given.

3.3. Obtaining Sink Position From the Ring

A source node, which has sensor data available, has to obtain the position of the current anchor node before disseminating data to the sink. Since the fresh position of the anchor node is stored in the ring, the source node has to obtain it from the ring. In order to do so, a similar mechanism to the delivery of ANPI packets to the ring is

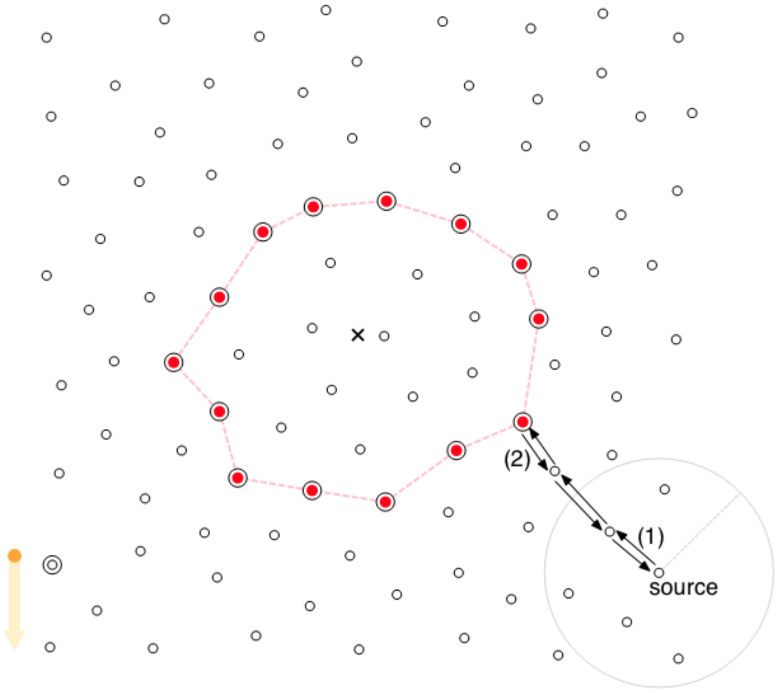


Figure 3.7. Anchor node position request (1) and response (2).

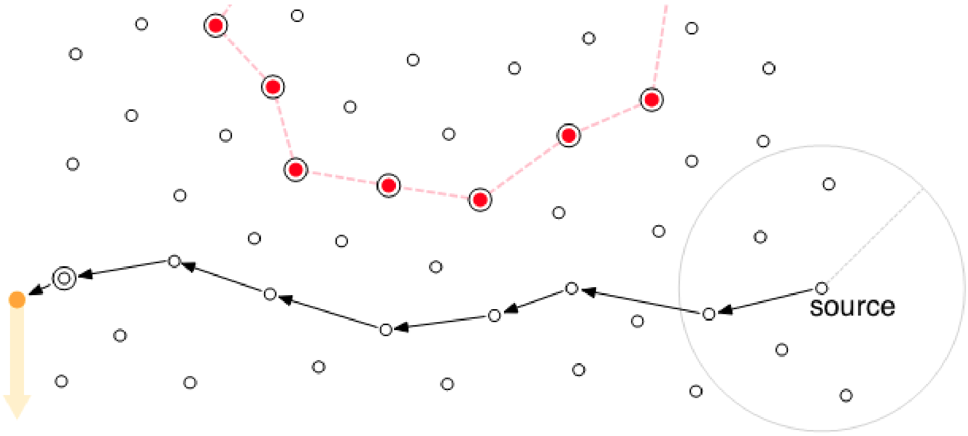


Figure 3.8. Data dissemination.

used. The source node sends an anchor node position information request (ANPIREQ) packet towards the ring (towards the network center if the node is outside the ring, away from the network center if it is inside the ring). It also includes its own position information in the ANPIREQ packet. The ring node receiving the ANPIREQ packet generates an anchor node position information response (ANPIRESP) packet which contains the current anchor node's position information and sends it to the source node making the request via geographic forwarding by using the position information of the source node retrieved from the ANPIREQ packet (Figure 3.7). Upon reception of the ANPIREQ packet, the source node now knows the position of the anchor node and can send its data towards the anchor node (Figure 3.8). The pseudo-codes for handling ANPIREQ and ANPIRESP packets are provided in Figure 3.9.

Another property of Ring Routing is that once a source node obtains the current anchor node's position, it remembers it for a specified amount of time determined by the anchor node history time (ANHT) parameter. This mechanism aims to reuse the anchor node position information for multiple data packets and decrease the frequency and thus the overhead of ANPI requests and responses.

There are several packet types which contain the anchor node position information (ANS, ANPI, ANPIRESP packets). Even though these packets and the information

```
procedure ProcessANPIREQ($ANPIREQ$)
    if $role = regularnode$ then
        if $ringnode \in neighbors$ then
            send ANPIreq to $ringnode$
        else
            $destinationPosition \leftarrow ANPIREQ.destinationPosition$
            GeographicRouting($ANPIREQ, destinationPosition$)
        end if
    else if $role = ringnode$ then
        create $ANPIRESP$
        set anchor node position and MAC address information in $ANPIRESP$
        $destinationPosition \leftarrow ANPIREQ.sourcePosition$
        GeographicRouting($ANPIRESP, destinationPosition$)
    end if
end procedure
procedure ProcessANPIRESP($ANPIRESP$)
    $destinationAddress \leftarrow ANPIRESP.destinationAddress$
    record anchor node position and MAC address information
    if $selfMACaddress = destinationAddress$ then
        for all $data \in queue$ do
            GeographicRouting($data, anchorNodePosition$)
        end for
    else if $destinationNode$ with $destinationAddress \in neighbors$ then
        send $ANPIRESP$ to $destinationNode$
    else
        $destinationPosition \leftarrow ANPIRESP.destinationPosition$
        GeographicRouting($ANPIRESP, destinationPosition$)
    end if
end procedure
```

Figure 3.9. Pseudo-code for processing ANPIreq and ANPIresp packets.

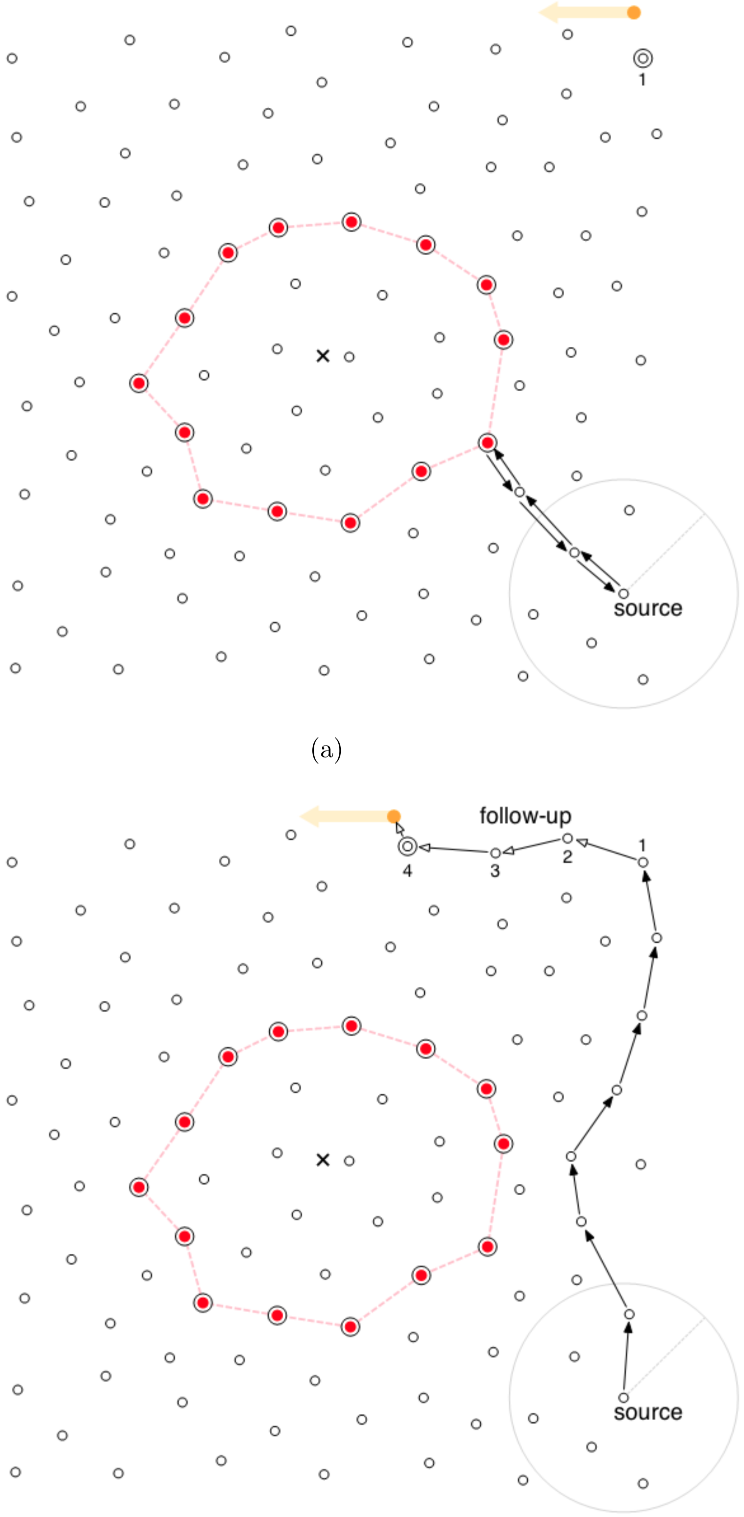
contained within them are destined for their ultimate target, the intermediate nodes relaying these packets may obtain the anchor node position information. Ring Routing uses this ability to its advantage, hence all intermediate nodes fetch anchor node position information from these packets and use it for delivering their own data to the anchor node whenever necessary. Fetching anchor node position information from the ANS broadcast also avoids unnecessary contact with the ring, since receiving an ANS broadcast means that the sink is very close, and direct communication is certainly the best way for data dissemination.

3.4. Data Dissemination

Once a source node receives a response (ANPIRESP) to its request (ANPIREQ), it learns the position of the anchor node and can now send its data directly to this anchor node by geographic forwarding (Figure 3.8). If data reaches an old anchor node, meaning that the anchor node has already changed by the time data has arrived at the destined anchor node, the follow-up mechanism is used to disseminate data to the current anchor node (Figure 3.10). The extra hops introduced by the follow-up mechanism is directly related to the ANHT parameter. Larger ANHT values will decrease the number of ANPI requests and responses; however, remembering the same anchor node position information for a long time will likely cause increased number of anchor node changes in the meantime. Each anchor node change means an extra hop introduced to the data dissemination path. Therefore, ANHT parameter has to be set carefully, while considering sink speed, to limit the number of extra hops.

3.5. Ring Change

Ring nodes are susceptible to consume more energy then other nodes since they process anchor node position information advertisements and requests. They handle more traffic than regular nodes. In order to prevent these nodes from dying quickly, they have to switch roles with regular nodes from time to time. We propose a simple mechanism to do that. The trigger of this mechanism might be a battery level



(a)

(b)

Figure 3.10. Follow-up mechanism.

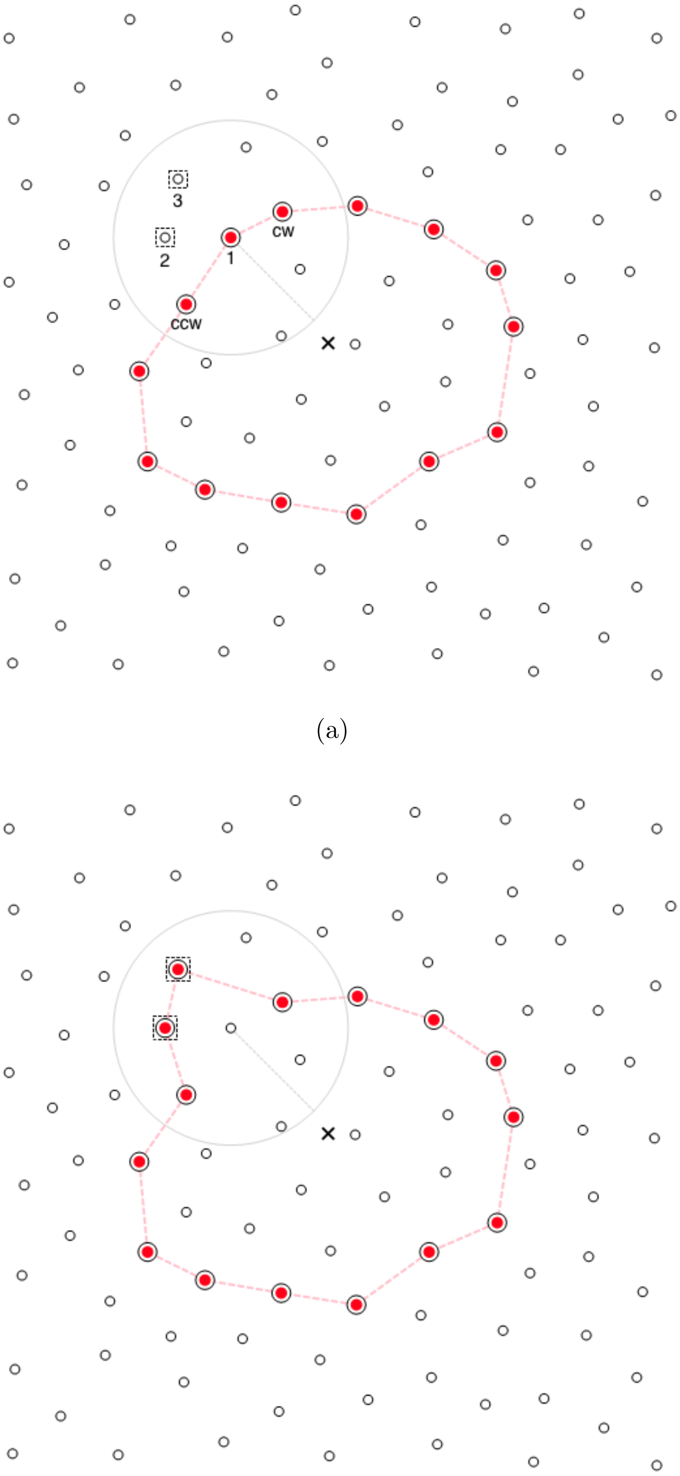
threshold, or it could simply be performed periodically.

Since there is no central control entity in a pure WSN architecture, the proposed ring change mechanism works locally, meaning that each ring node is independent to switch its role with regular nodes of its own choosing.

When a ring node decides to change its role to a regular node, it has to select ring node candidates (among regular nodes in its neighborhood) which will take on the role of ring nodes in its place. With the newly selected ring nodes, the two properties of the ring have to be preserved: encapsulation of the network center property and closed loop property. Also the ring node candidates have to be selected according to the current ring change direction (expanding or collapsing). In order to preserve the closed loop property of the ring, the ring candidates have to make a connection (single hop or multi-hop) between the ring node's clockwise and counter-clockwise ring neighbors. This could be achieved by running a shortest path algorithm on the search space of neighbors in the current ring change direction. After ring candidates are determined, a simple geometric check has to be done to ensure that network center is still encapsulated by the ring. After these two procedures are completed the ring node broadcasts a ring change (RC) packet which informs the ring candidates of their new role. At this point the ring node drops the role ring node and becomes a regular node. The last step is for the new ring nodes to inform their neighbors of their new ring node roles. Moreover, all regular nodes in the vicinity should update their position with respect to ring (inside/outside) information, in case the ring passes over them. This could be done by retrieving and using the ring change direction information available in the RC broadcast packet.

The pseudo-code of the procedure executed when a ring node decides to change its role is given in Figure 3.12.

The ring change mechanism is depicted in Figure 3.11. Node 1 attempts to switch its ring role, so it chooses two ring node candidates among its neighbors (node 1 and



(a)

(b)

Figure 3.11. Ring change (expanding).

node 2, marked with squares in Figure 3.11a). These two nodes are sufficient to preserve the closed loop property of the ring. Node 1 then broadcasts a RC packet informing these two new ring nodes, and the ring has successfully changed (Figure 3.11b).

If the ring node issuing a ring change operation is unable to find new ring candidates in the current ring change direction, the direction is simply reversed and the search is repeated. An example of this situation is when the ring reaches the edges of

```
procedure ChangeRing($ringChangeDirection$)
    $ringNodeCandidates \leftarrow$ DetermineRingNodeCandidates($ringChgDir$)
    $start \leftarrow ccwneighbor$
    $goal \leftarrow cwneighbor$
    $newRingNodes \leftarrow$ findShortestPath($start, goal, ringNodeCandidates$)
    assert that the $polygon$ formed by $newRingNodes$ does not
        encapsulate $network\ center$
    create $RCpacket$
    set $newRingNodes$ in $RCpacket$
    broadcast $RCpacket$
    $role \leftarrow regularnode$
end procedure
procedure DetermineRingNodeCandidates($ringChangeDirection$)
    for all $node \in neighbors$ do
        if $ringChangeDirection = Expand \wedge node$ is outside ring then
            $ringNodeCandidates \leftarrow ringNodeCandidates \cup \{node\}$
        else if $ringChangeDirection = Contract \wedge node$ is inside ring then
            $ringNodeCandidates \leftarrow ringNodeCandidates \cup \{node\}$
        end if
    end for
    return $ringNodeCandidates$
end procedure
```

Figure 3.12. Pseudo-code for ring change mechanism.

the network. In this case the ring stops expanding and begins to collapse until it can no longer collapse and it begins to expand again, and so on. If search in both directions fail, the change does not occur and the ring node continues to act as a ring node.

Since the ring change decisions are made locally, the shape of the ring might be imperfect in an instance of operation. However, imperfection of the ring would not impact correct operation as long as the closed loop and the encapsulation of the network center properties are met.

4. PERFORMANCE EVALUATION OF RING ROUTING

In order to evaluate the performance of Ring Routing, extensive simulations are conducted using the OPNET modeler environment [55]. In this section, the scenario and the parameters used for the simulations, results and the comparative performance evaluation of Ring Routing are presented. Furthermore, the details of the underlying MAC protocol used in OPNET simulations, X-MAC, and the extension we made to X-MAC to handle transmission of broadcast packets are explained. Even though Ring Routing does not have any MAC layer requirements, a MAC protocol is chosen and implemented in OPNET for the purposes of realistic simulations. X-MAC is a good candidate since it is shown to be an efficient low-power MAC protocol.

4.1. X-MAC Details and Extension

X-MAC [56] is basically the adaptation of low power listening (LPL) [57] for packetized radios. In order to understand the basics of X-MAC, first we need to explain how LPL works.

In LPL, when a node has a packet to send, it transmits an extended preamble. Nodes waking up sample the medium, and if they detect a preamble they remain awake for the duration of the preamble to determine if the packet is destined for themselves. After the preamble the sender transmits the data packet. The destination is included in the header; therefore, nodes receiving the header first could determine the destination node address. The nodes other than the destined node immediately go back to sleep.

LPL approach eliminates the need to establish synchronized sleep cycles; however, it creates extensive overhearing. The nodes have to remain awake for the whole duration of the preamble even if the data is not destined to themselves. X-MAC eliminates this problem by dividing the long preamble into smaller preamble (PRM) packets which contain the destination address. By this mechanism, the nodes could determine the destination much earlier and go back to sleep if the data packet is not meant to be

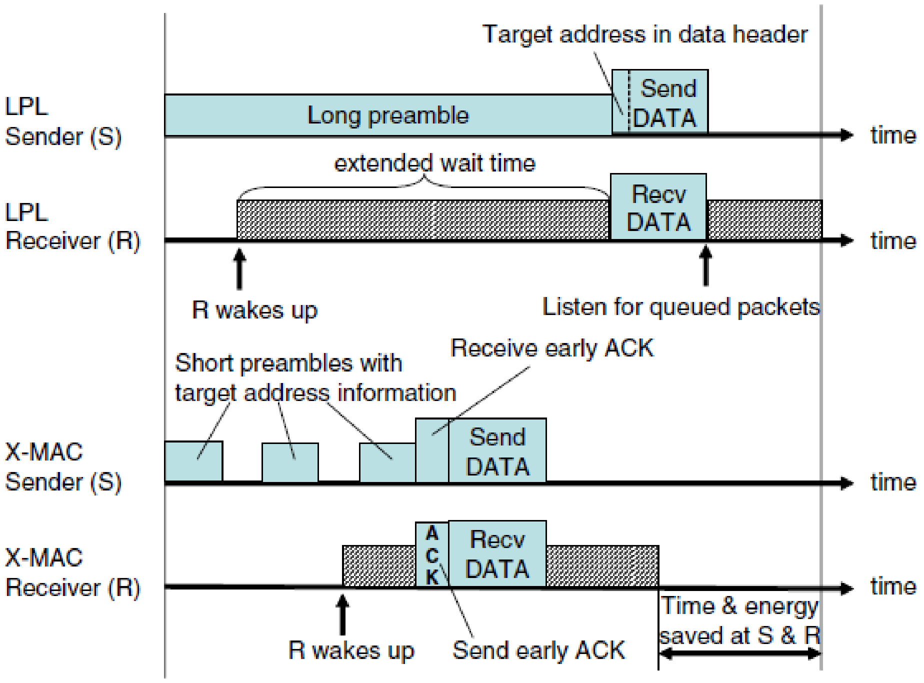


Figure 4.1. Comparison of the timelines of LPL and X-MAC [56].

sent to them. In order to ensure that the transmitter establishes a rendezvous with the receiver, the preamble period is at least as long as the maximum sleep period.

X-MAC further increases the utilization of the channel and decreases the duration of the active period by the pre-ack mechanism. The destination node hearing a preamble can immediately send a pre-ack in-between preambles to inform the sender that it is awake and ready to receive the data. A preamble pre-ack pair also acts similar to the RTS-CTS mechanism and is useful in providing collision avoidance.

In Figure 4.2, the difference between the long preamble approach of LPL and the strobed preamble approach of X-MAC is shown. The time and energy savings of such a strobed approach is clearly visible.

X-MAC also addresses the problem of multiple transmitters, which are waiting to send data to the same receiver, sending their entire preambles even though the receiver is awake. If a secondary node detects a preamble which is destined for the

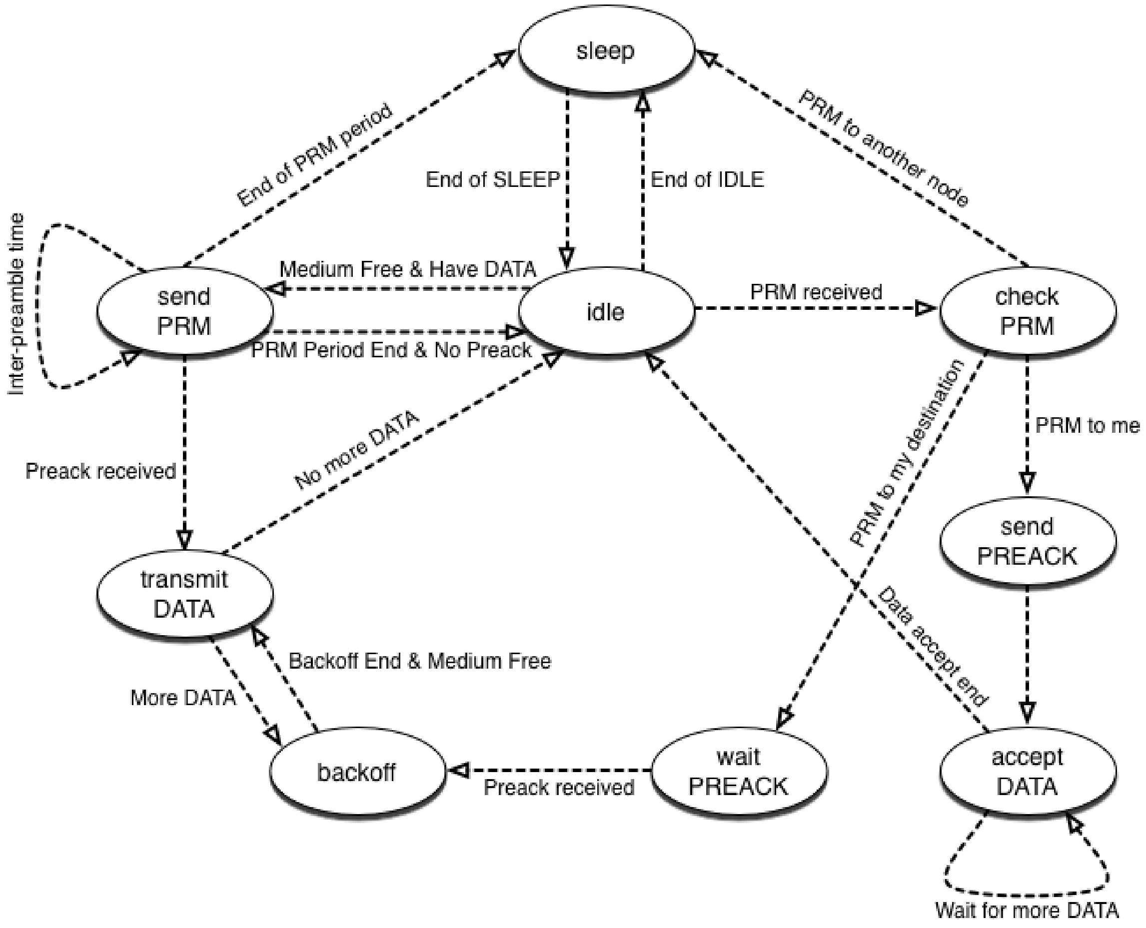


Figure 4.2. State transition diagram of X-MAC used in the OPNET implementation.

receiver node it has intentions to send a data packet to, it waits until the pre-ack arrives from the receiver node and then back-offs for a random period of time and attempts to send its packet without preambling. The random backoff is required to avoid collisions since more than one transmitter may be in a similar situation and be waiting to send packets. Multiple data packets may be transmitted in succession (with back-offs between data packets) using this mechanism. The preambling node may also participate in contention to send more data packets.

The authors of [56] did not provide implementation details for transmitting broadcast data packets. A naive approach would be to record a special global address in the preamble as the destination address which would be applicable to all receivers. Every node in the neighborhood of the transmitter would stay awake after hearing such a preamble and wait for the transmission of data. Pre-ack mechanism cannot be used in this case since there is no single eligible receiver, and every receiver in the neighborhood must be awake during a portion of the preamble period. This requirement necessitates that the preamble period is always as long as the maximum sleep time. This approach is not energy-efficient due to the excessive redundant preambles, and the necessity for all receivers to remain awake until the end of the preamble period.

Rather than the approach described above, we use a similar approach to BoX-MAC [58] to handle transmission of broadcast packets. BoX-MAC sends data packets rather than preambles during the preamble period. This approach is efficient for small data sizes but can be time and energy consuming for large data since a node would have to listen through a whole data packet to assess if the packet is destined to itself. However, in our application scenario broadcast packets contain only routing control information (e.g. ANS packet). This property allows us to use this approach since routing control information packets are likely to be small. When a receiver wakes up and receives a copy of the broadcast data, it can go immediately back to sleep to save energy. Tests conducted in OPNET show that this approach indeed is beneficial in terms of energy-efficiency and has a high reliability.

4.2. Performance Metrics

Energy is the most important resource in WSNs since sensor nodes' capabilities are sustained until their batteries are depleted. Performance of a WSN is directly related to its lifetime. We define lifetime as the time until a sensor node in the network dies due to the depletion of its battery [59]. We measure the energy consumption of each node in our experiments. Sensor nodes consume energy at different rates during transmission, reception, idle waiting and sleeping, which are separately calculated and summed up in the experiments.

Average energy consumption per node is determined to be the first energy-related performance metric. It reflects the overall energy consumption in the network. The second metric is determined to be the network lifetime which is increased by the degree of energy consumption uniformity across the network.

The data reporting delay is another important metric which effects the WSN performance. Especially for real-time application scenarios, the validity of the sensor data is directly related to its freshness. Reporting delays are largely affected by the overall traffic in the network and the number of hops data packets travel from the source to the sink. A portion of the overall traffic comes from the overhead of the routing protocol; therefore, a routing algorithm with high overhead would likely cause increased delays. Hop counts are influenced by the freshness of the sink position information since sending data packets to an outdated sink position would cause them to disseminate through longer paths (i.e., follow-up mechanism).

4.3. Scenario and Simulation Parameters

Although WSNs with mobile sinks are suitable for various application scenarios, a real-time periodic data reporting application such as a habitat monitoring system (sensor data consisting of temperature, humidity, etc.) is a strong candidate for the performance evaluation for two reasons. First, periodic data applications involve every node in the network to generate and report data, which is challenging due to the necessity for the routing protocol to serve the whole network efficiently. Second, real-time applications require minimization of reporting delays as well as energy consumption which poses another challenge. These two challenges render the real-time periodic data application scenario to be appropriate for the performance evaluation of Ring Routing.

A topology of 200 sensor nodes deployed uniform randomly on an area of $600 \times 600m^2$ is used for simulations. A single sink moves randomly in the area of deployment with constant speed. Experiments model 60 minutes of simulation time. On the other hand, the lifetime measuring test cases are as long as the the time spent until a node's battery is depleted. Each case is simulated 20 times with different seeds in order to

Table 4.1. Simulation Parameters.

Parameter	**Value**
Area of Deployment	$600 \times 600m^2$
Number of Nodes	200
Communication Range	$80m$
Data Packet Size	40 bytes
Reporting Frequency	$1/60s/node$
Buffer Size	20 packets
X-MAC: Sleep Time	$100ms$
X-MAC: Idle Time	$4ms$
Energy: Transmission	$17.4mA$
Energy: Reception	$19.7mA$
Energy: Idle	$20\mu A$
Energy: Sleep	$1\mu A$
Battery Capacity	$5000mJ$
Data Rate	250 Kpbs
Ring Routing: ANHT	$10, 70, 130$ s
LBDD: Line Width	$150m$
Railroad: Rail Width	$100m$
Sink Speed	$0, 3, 6, 9, 12, 15$ km/h
Number of Repetitions	20
Modeled Simulation Time	60 $min.$

eliminate the effect of the stochastic nature of some simulation events.

The default parameters used in our simulations (unless otherwise stated) are presented in Table 4.1. Energy expenditure values are for MICAz [60] motes.

Ring Routing's performance is compared with LBDD and Railroad in terms of the determined performance metrics, which are (i) average energy consumption per node, (ii) lifetime and (iii) average reporting delay. In all the cases the successful data delivery ratios are above 99% which proves the reliability of the evaluated protocols.

4.4. Simulation Results and Performance Evaluation

4.4.1. The Effect of the ANHT Parameter on Ring Routing's Performance

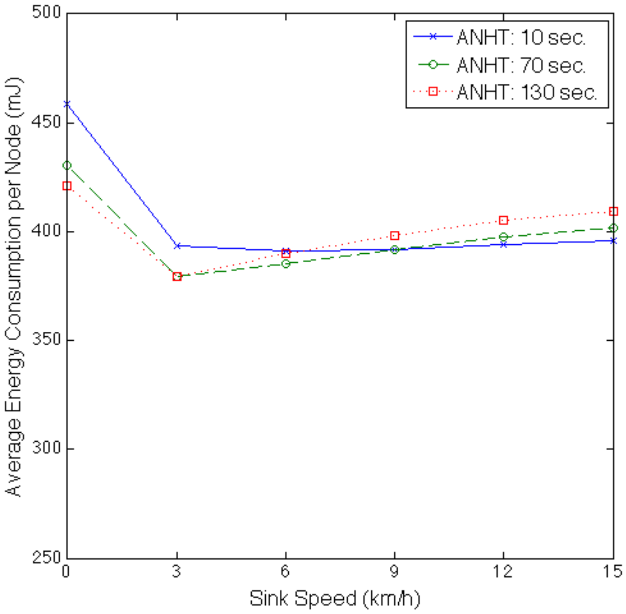
The ANHT (anchor node history time) parameter, which determines the time during which the acquired anchor node position is remembered, has a significant effect on the performance of Ring Routing. Therefore, we first evaluate Ring Routing with different ANHT values in order to determine the optimal ANHT values with respect to various sink speed, network size (number of sensor nodes and area size) and communication range values.

The sink speed affects the performance of a WSN with a mobile sink significantly. As the sink speed increases, the frequency of anchor node changes and anchor node position advertisements increase. However, slow sink motion is still susceptible to hotspots around the sink since the nodes in the vicinity of the sink rarely change. Due to these properties the performance of a WSN with a mobile sink largely depends on the speed of the sink.

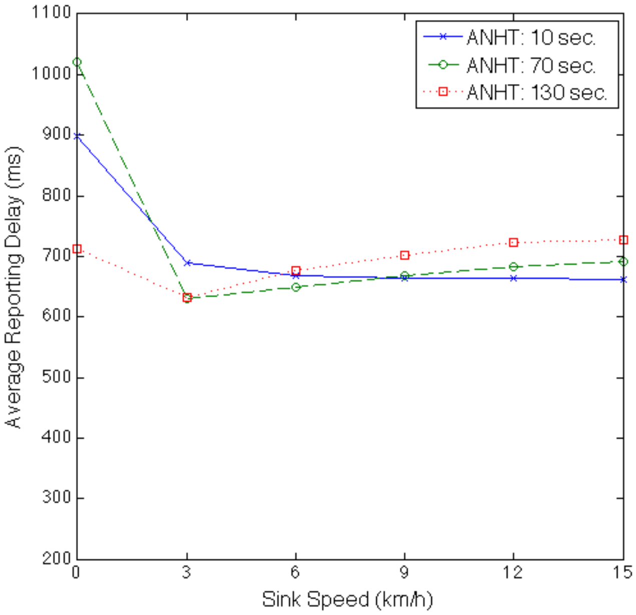
Figure 4.3 shows the average energy consumption per node across the network and the average reporting delay for three different ANHT values: 10 seconds, 70 seconds, 130 seconds. Given that each node generates and reports sensor data every 60 seconds, these three ANHT values correspond to sending one, two, three packets with the same acquired anchor node position information respectively.

In order to show the applicability of Ring Routing to the static sink WSNs, 0 km/h sink speed results are also included. In static sink cases, the sink resides in the northwest corner of the deployment area.

As seen in Figure 4.3a, larger ANHT values are suitable for slower sink speed values in terms of energy consumption. Since a slow sink rarely changes anchor nodes,



(a)



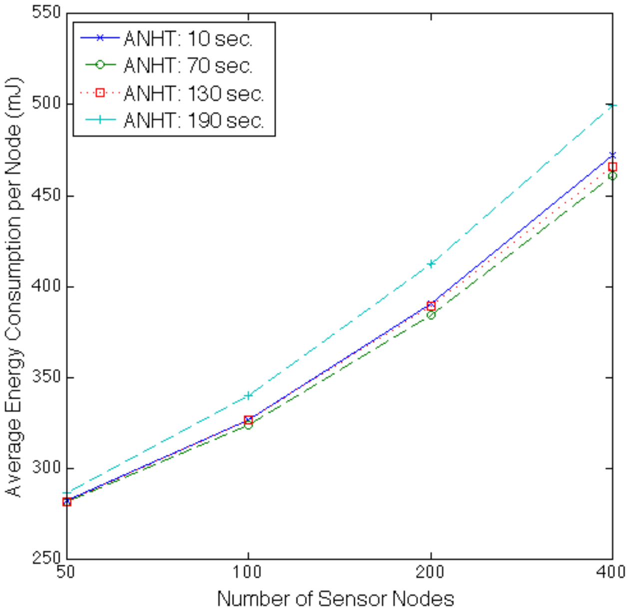
(b)

Figure 4.3. Effect of ANHT on (a) average energy consumption and (b) average reporting delay for changing sink speeds.

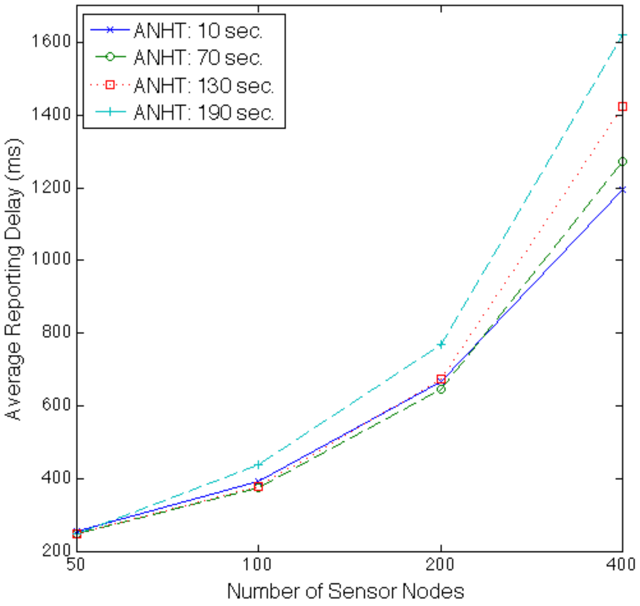
remembering the anchor node position longer is advantageous. As the sink speed increases, larger ANHT values are likely to cause extra hops introduced by the follow-up mechanism since anchor node changes are more frequent; therefore, the energy consumption of the network increases. For sink speeds of $3km/h$ and $6km/h$ the usage of 70 seconds ANHT has the lowest energy consumption. For larger sink speeds 10 seconds ANHT, which enforces the source nodes to send an ANPI request for each data packet, takes over.

The average reporting delays (Figure 4.3b) follow a similar pattern. The crossing points of the curves are also located around the same sink speed spots. The reason that the energy consumptions and the reporting delays are correlated is the fact that they are both affected by hop counts. The extra hops introduced by the follow-up mechanism in the high ANHT and high sink speed case also effect delays since delays are primarily affected by hop counts. On the other hand, for low ANHT values, the nodes frequently send ANPI requests to the ring which introduces extra traffic. Extra traffic increases the overall energy consumption, and the need to wait for a response before being able to send data to the anchor node residing in the acquired position introduces increased delays.

The sink speed is not the only parameter influencing the effect of the ANHT parameter on the WSN performance. ANHT introduces a trade-off between the costs of ANPI requests/responses and the extra hops of the follow-up mechanism. Since the severity of both these overheads are affected by packet hop counts, WSN parameters affecting the overall hop counts in the network influence the optimal ANHT values. The overall hop counts in a WSN are mainly determined by the network topology which is defined by the area of deployment, number of sensor nodes and communication range. In order to investigate the effect of these parameters, we first evaluate Ring Routing with different network sizes (defined by the area of deployment and the number of sensor nodes) while keeping the communication range and the average degree of the network (the average number of neighbors) fixed.



(a)



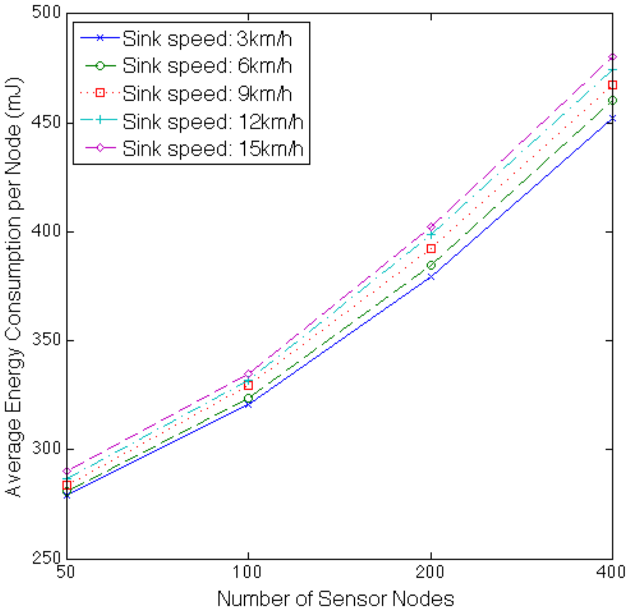
(b)

Figure 4.4. Effect of ANHT on (a) average energy consumption and (b) average reporting delay for various network sizes.

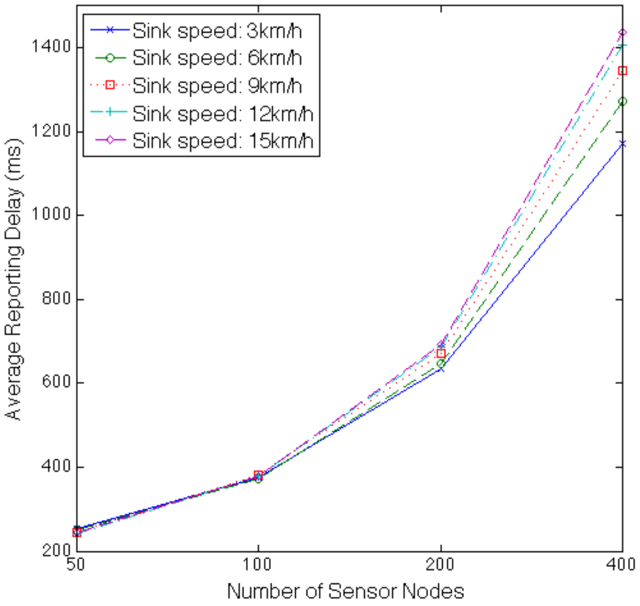
We define three topologies in addition to the standard 200 nodes topology ($600 \times 600m^2$ deployment area). The average degree of the standard topology (≈ 10) is maintained in all these additional topologies. We select numbers of sensor nodes to be 50, 100, 200 and 400. The respective deployment area sizes are $300 \times 300m^2$, $425 \times 425m^2$, $600 \times 600m^2$ and $850 \times 850m^2$. As the network gets larger, the expected packet hop counts increase.

The set of simulations performed with different network sizes use the sink speed of $6km/h$. This value is selected because the performances of three different ANHT values (10, 70 and 130 sec.), in terms of both energy consumptions and reporting delays, are close to each other in the $6km/h$ sink speed case. We wish to determine the differentiation of different ANHT curves with respect to different network sizes, and thus selecting such a sink speed value eliminates the effect of the sink speed as much as possible. Moreover, since the degree of the network and the sink speed is fixed, the rate of the anchor node changes (employed by the sink) also remains stationary in all of the cases.

Figure 4.4 shows the effect of ANHT on the average energy consumption and the average reporting delay for the four topologies defined above. One would expect higher ANHT values to have better performance for increased overall hop counts since the ratio of the extra hops introduced by the follow-up mechanism to the total number of data packet hops would decrease, thus rendering the overhead of the follow-up mechanism considerably low. However, this is not the case in Figure 4.4. The average energy consumption values of the three ANHT curves (10 sec., 70 sec., 130 sec.) in Figure 4.4a are close to each other. The average reporting delay curves (Figure 4.4b) are also close to each other with the exception of the 400 nodes case, in which the reverse of the expected behavior is observed: The higher ANHT values perform the worst. This unexpected pattern is due to the fact that an increase in the number of sensor nodes with the same reporting periods causes proportionally increased overall data traffic in the network. The data traffic is concentrated around the sink. Even though the frequent anchor node changes due to the sink's mobility tend to shift and mitigate



(a)



(b)

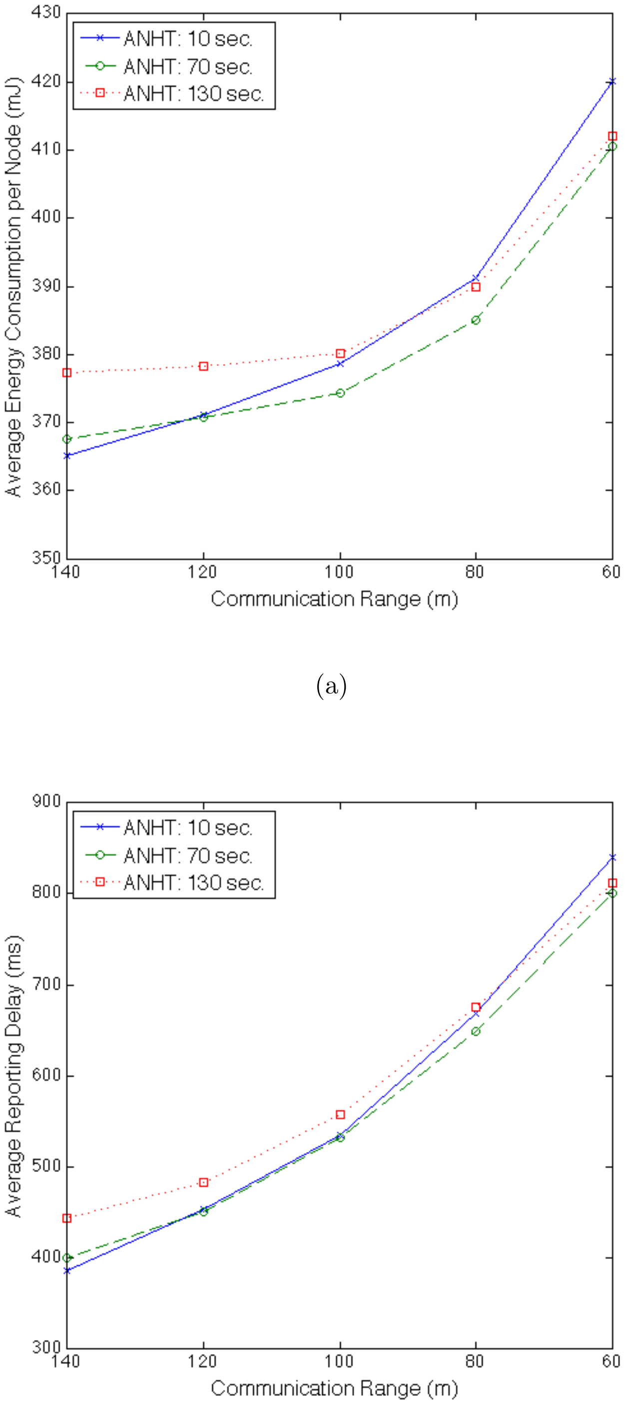
Figure 4.5. Effect of sink speed on (a) average energy consumption and (b) average reporting delay for various network sizes.

hotspots, the follow-up paths utilized by numerous source nodes for the duration of the ANHT are likely to become congested. For a high ANHT value, all nodes disseminate their data through the same long and rarely changed follow-up path. The nodes residing on this path handle increased data traffic and thus consume more energy, which also causes an increase in the overall energy consumption of the network since collisions and inefficiencies in the data transmissions are imminent.

The unexpected behavior in the reporting delays of the 400 nodes case (Figure 4.4b) is because the increased number of data packets traversing the follow-up path are likely to be buffered and forced to wait until the previously buffered packets are transmitted. The overall closeness of the different ANHT curves is due to the fact that the expected advantage of the higher ANHT values for higher overall hop counts and the disadvantage of the increased traffic on the follow-up paths caused by the increase in the number of nodes balance each other out.

Contrary to the closeness of the three ANHT curves (10 sec., 70 sec., 130 sec.) of average energy and average reporting delays, the largest ANHT (190 sec.) value clearly has the worst performance in all cases shown in Figure 4.4. This observation indicates that ANHT values larger than a certain threshold always pose an inefficiency since the disadvantage of utilizing excessively long follow-up paths outweighs the advantage of decreased frequency of ANPI request/responses.

The effect of sink speed on the average energy consumption and the average reporting delay for varying network sizes is shown in Figure 4.5. The fixed ANHT value of 70 sec. is used for this set of simulations. The results demonstrate the scalability of Ring Routing. For the 400 nodes case, which uses a significantly large topology, the average reporting delays and the average energy consumptions are within reasonable limits for the whole range of sink speed values, which proves that Ring Routing maintains its applicability to the real-time scenarios, and provides sufficiently good battery-conservation even for large topologies and high sink speeds.



(a)

(b)

Figure 4.6. Effect of ANHT on (a) average energy consumption and (b) average reporting delay for various communication ranges.

In order to be able to accurately observe the effect of the variation of overall hop counts on the optimal ANHT value, the load in the network must be kept stationary. In the next set of simulations, we achieve that by keeping the network size fixed and varying the communication range values. However, the communication range parameter affects the rate of the anchor node changes which must also be fixed since it also impacts the influence of the ANHT parameter on the WSN performance. For a certain sink speed value, an increase in the communication range causes the anchor node change rate to decrease since it would take longer for the sink to leave the communication range of an anchor node. Hence, the sink speed must be modified directly proportional to the communication range in order to achieve a constant expected anchor node change rate.

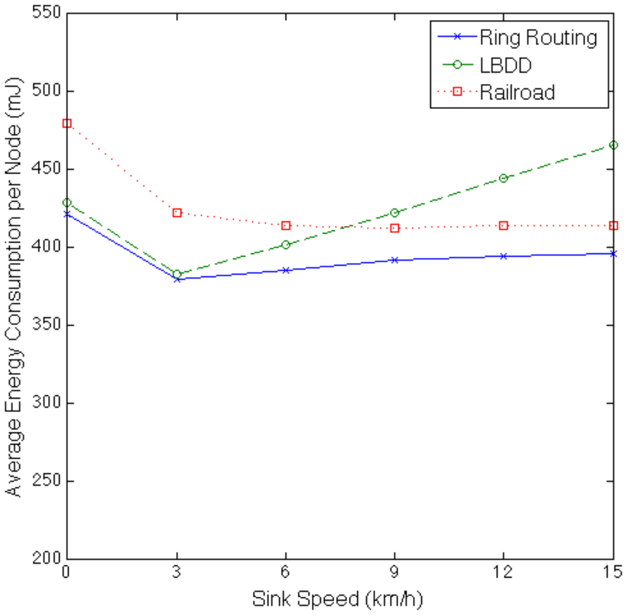
The corresponding sink speed values for the communication range values of 60, 80, 100, 120 and 140 meters are 2.25, 3, 3.75, 4.5 and 5.25 km/h respectively. Figure 4.6 shows the effect of ANHT on the average energy consumptions and the average reporting delays for different communication ranges. The x axes of the plots are in decreasing order since a decrease in the communication range values results in an increase in the expected packet hop counts.

We observe the expected behavior for the optimal ANHT values for varying communication range values in both the average energy consumption (Figure 4.6a) and the average reporting delay (Figure 4.6b) plots. For the large communication range value of 140 meters which corresponds to relatively lower expected packet hop counts, 10 sec. ANHT performs the best both energy-wise and delay-wise. For such a topology, the overhead of ANPI requests/responses is low and the extra hops introduced by the follow-up path in case of a large ANHT value would cause a significant overhead because the ratio of the length of the follow-up path to the length of the data dissemination path would be significantly larger compared to topologies with lower communication range values. As the communication range values decrease, the medium ANHT value of 70 sec. takes over and has the best performance for all communication range values. However, the performance of 130 sec. ANHT catches up even though it cannot outperform 70 sec. ANHT within the given communication range interval. We would expect its performance to get even better; however, it is very difficult to maintain the connectedness of the network for communication range values lower than 60 meters, and thus further evaluation was not possible.

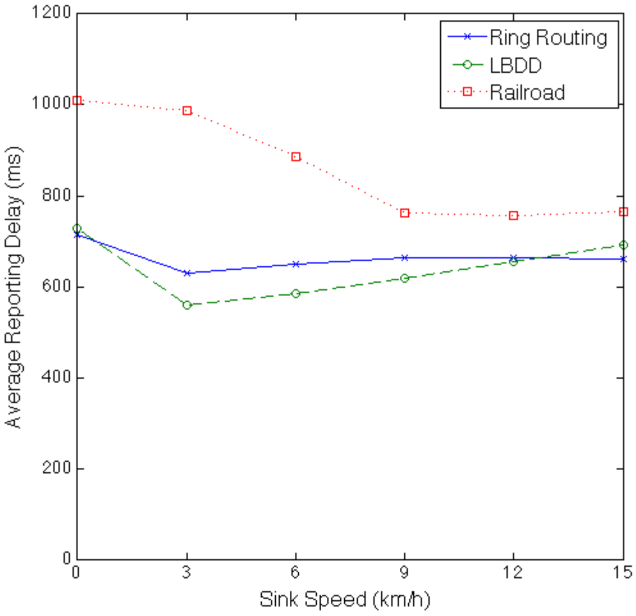
Having found the optimal values for the ANHT parameter under various conditions, we propose an adaptive ANHT selection mechanism. Since the effect of the ANHT parameter on the average energy consumption and the average reporting delays follows predictable patterns, the optimal ANHT value estimation can be extended beyond the measured conditions. The proposed mechanism works as follows: The sink, along with its position information, informs the newly selected anchor node of its current speed. The sink speed information is included in the ANPI packets and thus the ring is informed of the sink's current speed. Whenever a source node sends an ANPI request to a ring node, the ring node generating the response passes along the sink speed information. Using this information combined with the topological information (which can be loaded to the sensors' memories prior to the deployment), the source node sets its ANHT parameter adaptively to the estimated optimal ANHT value. In the comparative experiments (Section 4.4.2), Ring Routing uses the adaptive ANHT selection mechanism.

4.4.2. Comparative Performance Evaluation

Figure 4.7a shows the average energy consumptions of Ring Routing, LBDD and Railroad for varied sink speed values. Ring Routing has the best performance in all cases. LBDD performs better than Railroad for sink speed values $<\approx 8km/h$. LBDD's average energy consumption tends to increase monotonically. LBDD employs broadcasts along the line structure to share anchor node position information, thus increased rate of anchor node changes caused by increased sink speed leads to increased number of broadcast and thus elevated energy consumptions. Railroad limits the number of broadcasts along the rail by constructing localized stations in which the broadcasts are confined. The anchor node position information is shared by unicasts along the



(a)



(b)

Figure 4.7. Comparative performance: (a) average energy consumption and (b) average reporting delay for varying sink speeds.

rail until a station constructed by broadcasts is reached. Therefore, Railroad performs better than LBDD for faster sink speed values.

Ring Routing's energy-wise advantages do come at a cost. Figure 4.7b shows the average reporting delays of Ring Routing, LBDD and Railroad for different sink speeds. LBDD outperforms Ring Routing slightly in most cases while Railroad has the worst performance for all cases. This behavior is due to the ANPI request/response mechanism employed by Ring Routing and Railroad. LBDD sends data packets directly to the line which relays them to the sink, thus eliminating the delay cost of waiting for the response to an ANPI request. However, this approach is more susceptible to the hotspot problem, since the line nodes process larger amounts of data (bigger sized data packets rather than smaller sized requests). This trade-off is apparent in the energy consumption performance of LBDD which is worse than Ring Routing. Due to this very reason, we expect the lifetime performance of LBDD to be also weak.

In order to investigate the delay cost of the ANPI request/response mechanism

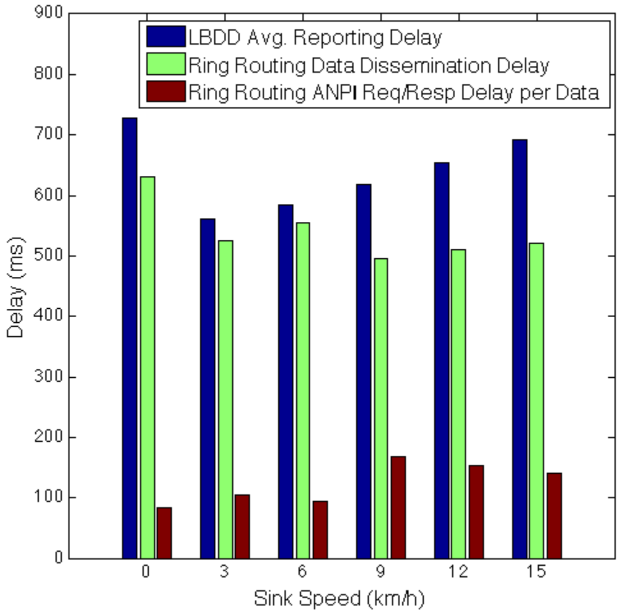
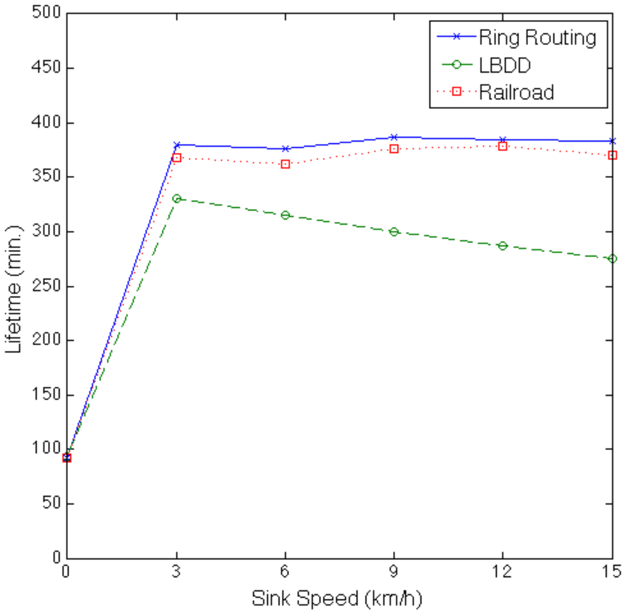


Figure 4.8. Delay breakdown of Ring Routing data delivery compared to LBDD.

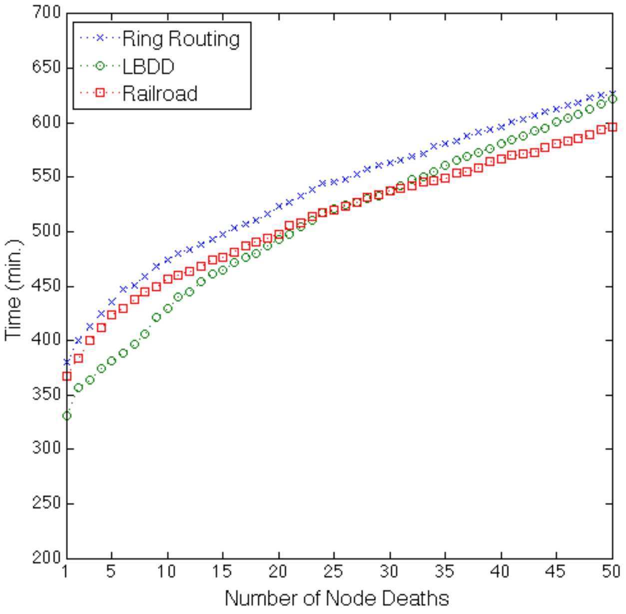
employed by Ring Routing, we provide the delay breakdown of the data delivery process of Ring Routing in Figure 4.8. The total delay for data deliveries are broken down into two components. The ANPI request/response delay per data component is the time until a response to the ANPI request is received by a source node. For the case of large ANHT values where the learned ANPI is used for multiple data packets, the delay of the request/response mechanism is divided to the number of data packets sent using the ANPI learned from the related response. The second component is the actual data dissemination delay of the path from the source to the sink.

The two components of Ring Routing's data delivery delays are compared with LBDD's average reporting delay. The delay cost of the request/response mechanism is apparent. The actual data dissemination delays of Ring Routing is lower than LBDD's reporting delays. Even though, discarding the request/response mechanism of Ring Routing and employing a direct data sending approach similar to LBDD would enhance the delay performance of Ring Routing, the energy advantages of the request/response mechanism outweighs the probable delay performance benefit. Since the delay values of Ring Routing is close to LBDD's delay values, and they are in reasonable limits to support real-time applications, the energy consumption performance is favored.

Even though the average energy consumption metric provides an insight to the projected longevity of the WSN operation, the lifetime metric is the clearer indicator. As a reminder, we define the lifetime of a WSN to be the time until the first node in the network dies. Figure 4.9a shows the lifetimes of Ring Routing, LBDD and Railroad for varying sink speed values. LBDD, as expected, has the worst lifetime performance among the three protocols due to the decreased degree of the energy consumption uniformity caused by all the data traffic being handled by the line nodes. The lifetimes of Railroad are close to but slightly worse than Ring Routing. Railroad employs a request/response mechanism similar to Ring Routing; however, broadcasts along the second-tier structure are not entirely avoided, thus enabling Ring Routing to stand out in terms of the network lifetimes. The static sink case is also provided to show the huge advantage of mobile sinks in terms of lifetime elongation.



(a)



(b)

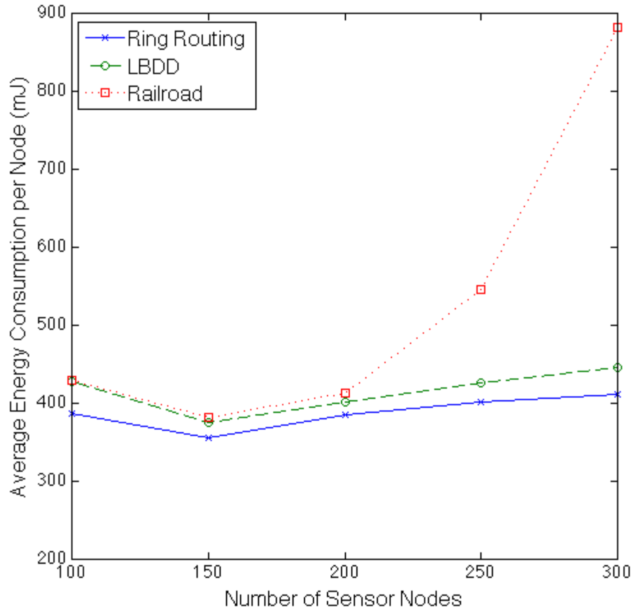
Figure 4.9. Comparative lifetime performance: (a) the time until the first node dies (b) death times for first 50 nodes.

The definition used for the network lifetime is reasonable but not enough to accurately assess the longevity of the WSN operation. The death of the first node might disrupt the topology and cause disconnectedness of some portions of the network, depending on the criticality of the dying node's position. However, these problems are implicitly mitigated by the usage of a mobile sink, since the sink is expected to eventually visit the disconnected areas in the network, thus providing stronger reliability and data delivery performance. Due to this property of the sink mobility, we must also evaluate the performance of the WSN until a significant portion of the network dies, in order to accurately and more realistically determine the lifetime of the network.

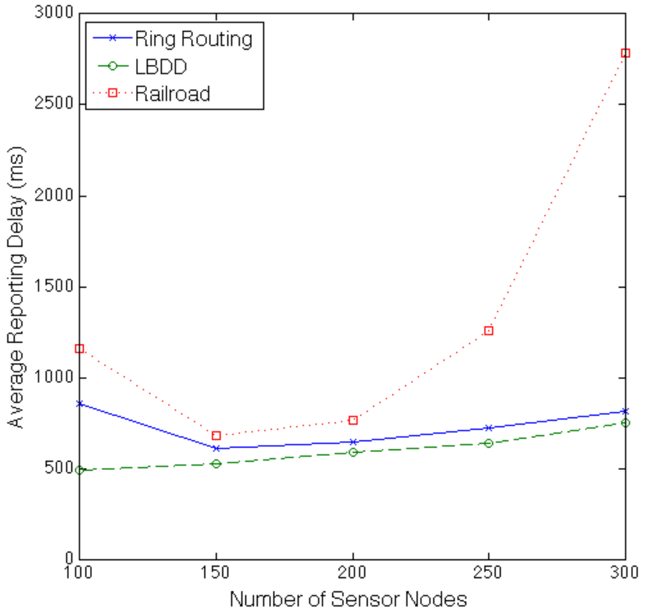
In Figure 4.9b, the distribution of the node death times until 25% (50 nodes) of the 200 node network dies is provided for the three protocols. The death of 25% of the network, which is a significant portion, hinders the WSN operation greatly, in terms of both the imminent topological disruptions and the decrease of the sensor field's coverage. The constant sink speed value of $3km/h$ is used for this set of simulations since the lifetime performances of the three protocols are observed to be the closest in the $3km/h$ sink mobility case.

Figure 4.9b emphasizes the strength of Ring Routing's lifetime performance. The deaths of the first 50 nodes in Ring Routing occur later than LBDD and Railroad on average. Moreover, for Railroad, the nodes die quicker than Ring Routing even though the first node death times are similar. LBDD's first node death times are much lower than Ring Routing and Railroad; however, the first 50 nodes death times distribution shows that LBDD's rate of node deaths decrease as more nodes die. LBDD performs better than Railroad after the death of $\approx 12\%$ of the network, due to the later node death times beyond this instance. LBDD nearly catches up with Ring Routing as the 25% of the network dies; however, evaluating beyond this value is irrational since the successful operation of the WSN is very likely to be already interrupted.

The sink speed is not the only parameter affecting WSN performance. The network size in terms of the number of deployed sensor nodes also affects WSN perfor-



(a)



(b)

Figure 4.10. Comparative performance: (a) average energy consumption and (b) average reporting delay for varying numbers of sensor nodes.

mance significantly since the density of the network and the total traffic loads depend on the network size. In the next set of simulations, we evaluate the performance of the protocols for topologies of different numbers of sensor nodes (100, 150, 200, 250, 300) deployed on a fixed area of $600 \times 600m^2$. Figure 4.10 shows the results of these simulations.

In Figure 4.10a, we observe that Ring Routing's average energy consumption performance is better than LBDD and Railroad for cases. Ring Routing and LBDD have stable performances over different network sizes; however, Railroad's performance is significantly impeded for larger networks. This is due to the station building mechanism of Railroad. For each data packet, a station which covers a sub-area of the rail is constructed by broadcasts. Increased number of nodes lead to increased number of generated data packets which causes more stations to be built and more broadcasts to be issued.

The average reporting delays of the protocols for varied network sizes (Figure 4.10b follow similar patterns to the average energy consumption curves: LBDD and Ring Routing perform stably across the specified number of nodes range, while Railroad's performance deteriorate as the size of the network grows. Ring Routing has slightly worse reporting delay values than LBDD due to the energy-delay trade-off explained before: Ring Routing, unlike LBDD, utilizes an ANPI request/response mechanism to decrease the traffic load on the ring structure which saves considerable energy at the cost of extra delays.

In both plots presented in Figure 4.10, we observe that the protocols perform slightly worse in the 100 nodes case. This is due to the sparseness of the 100 nodes network. In such a barely connected network, straight data paths are difficult to find due to the nodes' limited choices of next hop neighbors.

The comparative performance evaluation of Ring Routing proves that it indeed is an energy-efficient, lifetime enhancing routing protocol with minimal data reporting delays. It outperforms LBDD and Railroad which are stronger than most of the other existing two-tier protocols in terms of energy consumptions and data dissemination delays.

5. CONCLUSION

In this thesis, we reviewed the existing mobile sink routing protocols in detail and determined their benefits and drawbacks. Using our findings, we proposed Ring Routing, a new mobile sink routing protocol which combines the advantages of the existing protocols. Ring Routing defines a ring structure which is easily accessible and easily changeable. With these properties, data reporting delays are minimized and the hotspot problem is mitigated.

Ring Routing's core function is to deliver the sink's position to the ring structure while the source nodes access the ring to retrieve the sink position information. The mechanisms employed to achieve this functionality are explained in detail. Moreover, a local decision-based ring change mechanism is proposed and described.

The performance of Ring Routing is evaluated by the extensive simulations conducted in the OPNET modeler environment. To achieve realistic simulations, X-MAC is implemented in OPNET as the underlying MAC protocol. A wide range of different scenarios with varying network sizes and sink speed values are defined and used. The effect of the ANHT parameter is explored and an adaptive optimal ANHT selection mechanism is proposed. Comparative performance evaluation results of Ring Routing with two efficient mobile sink protocols, LBDD and Railroad, which are also implemented in OPNET, are provided. The results show that Ring Routing indeed is an energy-efficient protocol which extends the network lifetime. The reporting delays also confined in reasonable limits which proves that Ring Routing is suitable for real-time applications.

In the future, we want to modify Ring Routing to support multiple mobile sinks. Even though the current definition of Ring Routing is easily adaptable to operate with multiple mobile sinks by simply storing all the sinks' position information on the ring and generating ANPIRESP packets with the position information of the sink closest to the source node; this would merely be an adaptation without considering the benefit of using multiple sinks. A rational and efficient modification would utilize the advantages of multiple sinks with additional mechanisms and structures. Therefore, we wish to propose extensions and modifications to Ring Routing so that it operates even more efficiently with multiple mobile sinks. Such an approach is necessary to justify the extra cost of deploying additional sinks to the WSN.

In this thesis, the performance evaluation employs only constant speed sink mobility. In the future, we wish to explore different patterns such as nomadic or controlled sink mobility.

Another possible future work is to formulate an analytical approach to the adaptive ANHT selection mechanism. The current mechanism fails to be applicable for all possible WSN conditions since it relies on the performance evaluation results of the previously evaluated cases. An analytical solution to the problem of selecting the optimal ANHT values is possible due the predictable patterns of the effect of ANHT on the WSN performance.

REFERENCES

1. Jaichandran, R., A. Irudhayaraj and J. Raja, “Effective Strategies and Optimal Solutions for Hot Spot Problem in Wireless Sensor Networks (WSN)”, *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, pp. 389 –392, 2010.

2. Olariu, S. and I. Stojmenović, “Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting”, *IEEE INFOCOM*, pp. 1–12, Society Press, 2006.

3. Luo, J. and J.-P. Hubaux, “Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks”, *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 3, pp. 1735–1746, 2005.

4. Chatzigiannakis, I., A. Kinalis and S. Nikoletseas, “Efficient Data Propagation Strategies in Wireless Sensor Networks Using a Single Mobile Sink”, *Computer Communications*, Vol. 31, No. 5, pp. 896–914, 2008.

5. Goldenberg, D. K., J. Lin, A. S. Morse, B. E. Rosen and Y. R. Yang, “Towards Mobility as a Network Control Primitive”, *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc ’04, pp. 163–174, ACM, New York, NY, USA, 2004.

6. Di Francesco, M., S. K. Das and G. Anastasi, “Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey”, *ACM Transactions on Sensor Networks*, Vol. 8, No. 1, pp. 1–31, 2011.

7. Liang, W., J. Luo and X. Xu, “Prolonging Network Lifetime via a Controlled Mobile Sink in Wireless Sensor Networks”, *Global Telecommunications Conference (GLOBECOM 2010), IEEE*, pp. 1 –6, 2010.

8. Wang, Z., S. Basagni, E. Melachrinoudis and C. Petrioli, “Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime”, *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS ’05)*, p. 287, 2005.

9. Rao, J. and S. Biswas, “Network-Assisted Sink navigation for Distributed Data Gathering: Stability and Delay-Energy Trade-offs”, *Computer Communications*, Vol. 33, No. 2, pp. 160–175, 2010.

10. Hamida, E. and G. Chelius, “Strategies for Data Dissemination to Mobile Sinks in Wireless Sensor Networks”, *Wireless Communications, IEEE*, Vol. 15, No. 6, pp. 31 –37, december 2008.

11. Faheem, Y., S. Boudjit and K. Chen, “Data Dissemination Strategies in Mobile Sink Wireless Sensor Networks: A Survey”, *Wireless Days (WD), 2009 2nd IFIP*, pp. 1 –6, 2009.

12. Grammalidis, N., E. Cetin, K. Dimitropoulos, F. Tsalakanidou, K. Kose, O. Gunay, B. Gouverneur, E. K. D. Torri, S. Tozzi, A. Benazza, F. Chaabane, B. Kosucu and C. Ersoy, “A Multi-Sensor Network for the Protection of Cultural Heritage”, *19th European Signal Processing Conference (EUSIPCO2011), Special Session on Signal processing for disaster management and prevention*, 2011.

13. Yun, Y. and Y. Xia, “Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications”, *Mobile Computing, IEEE Transactions on*, Vol. 9, No. 9, pp. 1308 –1318, 2010.

14. Shen, C.-C., C. Srisathapornphat and C. Jaikaeo, “Sensor Information Networking Architecture and Applications”, *Personal Communications, IEEE*, Vol. 8, No. 4, pp. 52–59, 2001.

15. Langendoen, K. and A. Meier, “Analyzing MAC Protocols for Low Data-Rate Applications”, *ACM Transactions on Sensor Networks*, Vol. 7, No. 2, pp. 1–40, 2010.

16. Al-Karaki, J. and A. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey”, *Wireless Communications, IEEE*, Vol. 11, No. 6, pp. 6–28, 2004.

17. Akkaya, K. and M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks”, *Ad Hoc Networks*, Vol. 3, No. 3, pp. 325–349, 2005.

18. Li, X., J. Yang, A. Nayak and I. Stojmenovic, “Localized Geographic Routing to a Mobile Sink with Guaranteed Delivery in Sensor Networks”, *IEEE Journal of Selected Areas of Communication*, 2012, to appear.

19. Karp, B. and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom ’00, pp. 243–254, ACM, New York, NY, USA, 2000.

20. Ma, C., L. Wang, J. Xu, Z. Qin, M. Zhu and L. Shu, “A Geographic Routing Algorithm in Duty-Cycled Sensor Networks with Mobile Sinks”, *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*, pp. 343 –344, 2011.

21. Ben Hamida, E. and G. Chelius, “A Line-Based Data Dissemination Protocol for Wireless Sensor Networks with Mobile Sink”, *Communications, 2008. ICC ’08. IEEE International Conference on*, pp. 2201–2205, 2008.

22. Shin, J.-H., J. Kim, K. Park and D. Park, “Railroad: Virtual Infrastructure for Data Dissemination in Wireless Sensor Networks”, *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. PE-WASUN ’05*, pp. 168–174, 2005.

23. Intanagonwiwat, C., R. Govindan, D. Estrin, J. Heidemann and F. Silva, “Directed Diffusion for Wireless Sensor Networking”, *Networking, IEEE/ACM Transactions on*, Vol. 11, No. 1, pp. 2–16, 2003.

24. Dai, H. and R. Han, “A Node-Centric Load Balancing Algorithm for Wireless Sensor Networks”, *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*, Vol. 1, pp. 548–552, 2003.

25. Fyffe, M., M.-T. Sun and X. Ma, “Traffic-Adapted Load Balancing in Sensor Networks Employing Geographic Routing”, *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 4389–4394, 2007.

26. Casari, P., M. Nati, C. Petrioli and M. Zorzi, “ALBA: An Adaptive Load - Balanced Algorithm for Geographic Forwarding in Wireless Sensor Networks”, *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1 –9, 2006.

27. Toumpis, S. and S. Gitzenis, “Load Balancing in Wireless Sensor Networks using Kirchhoff’s Voltage Law”, *INFOCOM 2009, IEEE*, pp. 1656 –1664, 2009.

28. Luo, H., F. Ye, J. Cheng, S. Lu and L. Zhang, “TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks”, *Wireless Networks*, Vol. 11, pp. 161–175, 2005.

29. Kweon, K., H. Ghim, J. Hong and H. Yoon, “Grid-Based Energy-Efficient Routing from Multiple Sources to Multiple Mobile Sinks in Wireless Sensor Networks”, *Wireless Pervasive Computing, 2009. ISWPC 2009. 4th International Symposium on*, pp. 1 –5, 2009.

30. Shon, M., C. Kong and H. Choo, “Hexagonal Path Data Dissemination for Energy Efficiency in Wireless Sensor Networks”, *Information Networking, 2009. ICOIN 2009. International Conference on*, pp. 1 –5, 2009.

31. Lin, C.-J., P.-L. Chou and C.-F. Chou, “HCDD: Hierarchical Cluster-Based Data Dissemination in Wireless Sensor Networks with Mobile Sink”, *Proceedings of the 2006 international conference on Wireless communications and mobile computing. IWCMC ’06*, pp. 1189–1194, 2006.

32. Kim, H. S., T. F. Abdelzaher and W. H. Kwon, “Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks”, *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys ’03, pp. 193–204, ACM, New York, NY, USA, 2003.

33. Mir, Z. and Y.-B. Ko, “A Quadtree-Based Data Dissemination Protocol for Wireless Sensor Networks with Mobile Sinks”, P. Cuenca and L. Orozco-Barbosa (Editors), *Personal Wireless Communications*, Vol. 4217 of *Lecture Notes in Computer Science*, pp. 447–458, Springer Berlin / Heidelberg, 2006.

34. Ratnasamy, S., B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, “GHT: A Geographic Hash Table for Data-Centric Storage”, *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA ’02, pp. 78–87, ACM, New York, NY, USA, 2002.

35. Lu, J.-L. and F. Valois, “On the Data Dissemination in WSNs”, *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, p. 58, 2007.

36. Fodor, K. and A. Vidács, “Efficient Routing to Mobile Sinks in Wireless Sensor Networks”, *Proceedings of the 3rd international conference on Wireless internet*, WICON ’07, pp. 1–7, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2007.

37. Xu, X., W. Liang and T. Wark, “Data Quality Maximization in Sensor Networks with a Mobile Sink”, *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pp. 1 –8, 2011.

38. Chen, Z., S. Liu and J. Huang, “Multi-Tier Grid Routing to Mobile Sink in Large-Scale Wireless Sensor Networks”, *Journal of Networks*, Vol. 6, No. 5, 2011.

39. Xun-Xin, Y. and Z. Rui-Hua, “An Energy-Efficient Mobile Sink Routing Algorithm for Wireless Sensor Networks”, *Wireless Communications, Networking and Mobile*

Computing (WiCOM), 2011 7th International Conference on, pp. 1 –4, 2011.

40. Heinzelman, W. R., A. Chandrakasan and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Vol. 8 of *HICSS ’00*, pp. 8020–8030, IEEE Computer Society, Washington, DC, USA, 2000.

41. Lee, E., S. Park, S. Oh, S.-H. Kim and K.-D. Nam, “Real-Time Routing Protocol Based on Expect Grids for Mobile Sinks in Wireless Sensor Networks”, *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pp. 1 –5, 2011.

42. Park, S., E. Lee, H. Park, J. Jung and S.-H. Kim, “Strategy for Real-Time Data Dissemination to Mobile Sinks in Wireless Sensor Networks”, *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pp. 1905 –1910, 2010.

43. Das, S., H. Pucha and Y. Hu, “Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing”, *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 2, pp. 1228 – 1239 vol. 2, 2005.

44. Shi, G., J. Zheng, J. Yang and Z. Zhao, “Double-Blind Data Discovery Using Double Cross for Large-Scale Wireless Sensor Networks With Mobile Sinks”, *Vehicular Technology, IEEE Transactions on*, Vol. 61, No. 5, pp. 2294 –2304, 2012.

45. Braginsky, D. and D. Estrin, “Rumor Routing Algorithm for Sensor Networks”, *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA ’02, pp. 22–31, ACM, New York, NY, USA, 2002.

46. Shi, L., B. Zhang, H. T. Mouftah and J. Ma, “DDRP: An Efficient Data-Driven Routing Protocol for Wireless Sensor Networks with Mobile Sinks”, *International Journal of Communication Systems*, 2012.

47. Ye, F., G. Zhong, S. Lu and L. Zhang, “GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks”, *Wireless Networks*, Vol. 11, pp. 285–298, 2005.

48. Lee, J., J. Kim, B. Jang and E.-S. Lee, “Data Dissemination Protocol Based on Home Agent and Access Node for Mobile Sink in Mobile Wireless Sensor Networks”, G. Lee, D. Howard and D. Slezak (Editors), *Convergence and Hybrid Information Technology*, Vol. 6935 of *Lecture Notes in Computer Science*, pp. 306–314, Springer Berlin / Heidelberg, 2011.

49. Wohlers, R., N. Trigoni, R. Zhang and S. Ellwood, “TwinRoute: Energy-Efficient Data Collection in Fixed Sensor Networks with Mobile Sinks”, *Mobile Data Management: Systems, Services and Middleware, 2009. MDM ’09. Tenth International Conference on*, pp. 192 –201, 2009.

50. Wang, G., T. Wang, W. Jia, M. Guo and J. Li, “Adaptive Location Updates for Mobile Sinks in Wireless Sensor Networks”, *The Journal of Supercomputing*, Vol. 47, pp. 127–145, 2009.

51. Safdar, V., F. Bashir, Z. Hamid, H. Afzal and J. Y. Pyun, “A Hybrid Routing Protocol for Wireless Sensor Networks with Mobile Sinks”, *Wireless and Pervasive Computing (ISWPC), 2012 7th International Symposium on*, pp. 1 –5, 2012.

52. Niculescu, D., “Positioning in Ad Hoc Sensor Networks”, *Network, IEEE*, Vol. 18, No. 4, pp. 24–29, 2004.

53. Albowicz, J., A. Chen and L. Zhang, “Recursive Position Estimation in Sensor Networks”, *Network Protocols, 2001. Ninth International Conference on*, pp. 35 –41, 2001.

54. Gopakumar, A. and L. Jacob, “Localization in Wireless Sensor Networks Using Particle Swarm Optimization”, *Wireless, Mobile and Multimedia Networks, 2008. IET International Conference on*, pp. 227 –230, 2008.

55. *OPNET Modeler 14.5.A PL8*, 2008, http://www.opnet.com/products/modeler/home.htm, Accessed at August 2012.

56. Buettner, M., G. V. Yee, E. Anderson and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks", *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, pp. 307–320, ACM, New York, NY, USA, 2006.

57. Hill, J. and D. Culler, "Mica: A Wireless Platform for Deeply Embedded Networks", *Micro, IEEE*, Vol. 22, No. 6, pp. 12 – 24, 2002.

58. Moss, D. and P. Levis, *BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking*, Tech. rep., SING-08-00, Stanford University, 2008.

59. Kang, I. and R. Poovendran, "Maximizing Static Network Lifetime of Wireless Broadcast Ad Hoc Networks", *Communications, 2003. ICC '03. IEEE International Conference on*, Vol. 3, pp. 2256 – 2261 vol.3, 2003.

60. Crossbow Technology, Inc., "MICAz Wireless Measurement System", Document Part Number: 6020-0060-04 Rev A, http://www.xbow.com, Accessed at August 2012.