ANT-COLONY OPTIMIZATION BASED IN-NETWORK DATA AGGREGATION IN

WIRELESS SENSOR NETWORKS

THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for the Degree

Master of SCIENCE

by

Meng Xie, M.S.

San Marcos, Texas
December 2011

IMPROVING ANT-COLONY OPTIMIZATION BASED IN-NETWORK DATA AGGREGATION IN WIRELESS SENSOR NETWORKS

Committee Members Approved:

Hongchi Shi, Chair

Xiao Chen

Qijun Gu

Approved:

J. Michael Willoughby
Dean of the Graduate College

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor and committee chair Dr. Hongchi Shi for his support, guidance, and insightful advice on my research and thesis. He has been a great advisor and friend to me.

I am also grateful to the rest of my committee members for their encouragement and comments. My special thanks go to Ms. Trish Sumbera for her kind help when I worked as a lab assistant with her. I would like to thank my lab mates for their insights, discussions, and jokes in the past two years. The experience was full of joy and enlightening.

I thank my parents for their support. My husband, Bolian, has been my pillar along the way. I could not come this far without his encouragement.

This manuscript was submitted on 11/7/2011 for final review.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABSTRACT

ANT-COLONY OPTIMIZATION BASED IN-NETWORK DATA AGGREGATION IN WIRELESS SENSOR NETWORKS

by

Meng Xie, M.S.

Texas State University-San Marcos

December 2011

SUPERVISING PROFESSOR: HONGCHI SHI

In-network data aggregation is an important technique in wireless sensor networks. It improves energy efficiency and alleviates congestive routing traffic by eliminating data redundancy in message passing processes. Ant-colony aggregation is a distributed algorithm that provides an intrinsic way of exploring search space to optimize settings for optimal data aggregation. This thesis aims to refine the heuristic function and the aggregation node selection method to maximize energy efficiency and extend network lifetime.

Introduction

Wireless Sensor Networks

A Wireless Sensor Network (WSN) (Pottie, & Kaiser, 2000; Akyildiz, Su, Sankarasubramaniam, & Cyirci, 2002; "Wireless sensor network", 2010) is a special *ad hoc* network spatially deployed with a large number of autonomous nodes equipped with sensors to cooperatively monitor physical or environmental conditions where unattended operation is required. Each sensor node is capable of sensing, computing, routing and communicating with other nodes or with the base station(s) (Al-Karaki, & Kamal, 2004; Fasolo, Rossi, Widmer, & Zorzi, 2007). A base station, also called sink node, is a fixed or mobile node used for connecting the sensor network to an existing communication infrastructure or the Internet where the user can access sensed data (Al-Karaki, & Kamal, 2004). WSNs are designed to be applied in industries such as transportation, manufacture, health care, environmental oversight, safety and security (Pottie, & Kaiser, 2000). Sensor nodes have to be low cost, power efficient for easy deployment in large quantities. Typical application environments such as monitoring, tracking, and surveillance along with restrained resource characteristics of sensor nodes lead to different network requirements and communication protocol designs for wireless sensor networks (Galluccio, Palazzo, & Campbell, 2009).

In-network Data Aggregation

For sensor network applications, in-network data aggregation and management allows trade-off between communication complexity and computation complexity (Fasolo, Rossi, Widmer, & Zorzi, 2007). Sensor network applications have the following attributes: 1. High data redundancy due to correlation; 2. *funneling effect* (Ahn, Hong, Miluzzo, Campbell, & Cuomo, 2006) - the closer a node is to the destination, the more demand for energy consumption and traffic congestion; 3. interest in only summarized information for some applications. By taking those attributes into consideration, data aggregation techniques manage to reduce and balance energy consumption (Intanagonwiwat, Govindan, Estrin, Heidenmann, & Siva, 2002; Boulis, Ganeriwal, & Srivastava, 2003) and alleviate traffic congestion (Galluccio, Campbell, & Palazzo, 2005). Fasolo, Rossi, Widmer, & Zorzi, 2007) define the in-network aggregation process as follows: *In-network aggregation is the global process of gathering and routing information through a multihop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular, energy), thereby increasing network lifetime.* Aggregation may or may not have size reduction, depending on the application purpose. Components of in-network aggregation includes: *routing protocols, aggregation functions,* and *ways of representing the data* (Fasolo, Rossi, Widmer, & Zorzi, 2007).

Data compression and data fusion techniques are also used to save energy and reduce traffic in WSNs. Although the concept is used interchangeably with data aggregation in some papers, they are defined differently. Data compression is for efficiently collecting all data observed by the sensor nodes using complex algorithms, though sometimes the terms of "compression" and "aggregation" are used interchangeably. Usually, it compresses the content at the original sources in a distributed manner without explicit routing-based aggregation (Deshpande, 2009). However, since sensor nodes are typically computationally constrained and have limited memories, it may not be feasible to run sophisticated data compression algorithms on WSNs (Kimura, & Latifi, 2005; Deshpande, 2009). The information theory of entropy is usually applied in data compression algorithms.

Data fusion in sensor networks is "the set of algorithms, processes, and protocols that combine data from multiple sensors" to extract information, "improve the quality of information compared to that provided by any individual data, or improve the operation of the network by optimizing usage of its resources" (Kansal, & Zhao, 2009). Sensed data from one or more events may be combined to help make decisions and "leverage past observations from the same sensors, previously known models of the sensed phenomenon, and other information in addition to the sensor data" (Kansal, & Zhao, 2009).

Networking Protocols for In-network Data Aggregation

At the network layer of the protocol stack, it is critical that energy-efficient routing methods are employed for relaying data to the sink node in order to maximize the network lifetime. The class of routing used for aggregation is also called *cooperative routing* (Al-Karaki, & Kamal, 2004). Different from classical *address-centric* routing protocols, routing of in-network aggregation is *data centric*. Nodes route packets based on the content of the packet instead of destination address. Routing approaches for in-network data aggregation are not only about the paths to forward data, but also about when, where and how. Timing or synchronization strategy is required. Decisions may need to be made upon whether to aggregate or not at a specific node, and ways to disseminate queries also vary in different routing protocols. Due to the significance of routing, extensive studies have been done, and numerous protocols have been proposed.

Routing schemes across a range of spatial-correlated WSNs were proposed by Pattem, Krishnamachari, and Govindan (2004): 1. Distributed Source Coding (DSC): Sensor nodes have perfect knowledge about data correlations, therefore, data is compressed or encoded before transmitting. Each node sends the data to the sink using the shortest path. 2. Routing Driven Compression (RDC): Sensor nodes do not know about the correlations between the data and send the data to the sink using the shortest path while allowing opportunistic aggregation when paths overlap. 3. Compression Driven Routing (CDR): Nodes also do not know about data correlations. Paths are initially routed, and aggregation is performed at selected hops. Falling into the CDR class are the four fundamental routing protocols based on the network structure: T*ree-based*, *cluster-based*, *multi-path,* and *hybrid*.

Aggregation Functions

An aggregation function is the concise algorithm performed on each aggregation node to compress or summarize information in order to reduce communication energy and meanwhile meet the purpose of the application. Aggregation function should be implementable by means of elementary operations because complex algorithms will consume too much computation power (Fasolo, Rossi, Widmer, & Zorzi, 2007). Besides, different devices may be suitable for different types of operations depending on energy sources and computation capabilities. Common paradigms of aggregation functions include *lossiness, duplicate sensitivity, exemplary or summary, monotonic aggregation* and *partial state results.* The last four paradigms were proposed by Madden, Franklin, Hellerstein, and Hong (2002).

Lossy or lossless. Data aggregation operation will result in either lossy or lossless information at the sink. Although only redundant data is compressed at each aggregation point, the process may cause loss in precision with respect to transmitting all readings uncompressed (Fasolo, Rossi, Widmer, & Zorzi, 2007). In lossy aggretaion, several packets are combined into a single packet, and thus the amount of outgoing data is significantly smaller than the input. This approach is useful when communication load exceeds system capacity and energy saving is needed. Therefore, there is a trade-off between energy efficiency and data precision. The degree of aggregation (DoA) is defined as the ratio of number of bits in all the packets considered for aggregation in one round and the number of bits in the aggregated packet (Padmanabh, & Vuppala, 2009). On the other hand, lossless approach compresses data by preserving its original information, therefore, readings can be perfectly reconstructed at the receiver side (Fasolo, Rossi, Widmer, & Zorzi, 2007). More information is embedded into a single packet by concatenating individual data items into larger packets, thus amortizing per-packet protocol overhead (Abdelzaher, He, & Stankovic, 2004). Lossless approach is effective when individual sensor readings are small in size so plenty room is available for concatenation in each packet.

Duplicate sensitive or duplicate insensitive. When a node receives multiple copies of identical information, it may process the data by taking into account the number of copies (duplicate sensitive) or considering one single copy of the data (duplicate insensitive) (Madden, Franklin, Hellerstein, & Hong, 2002; Fasolo, Rossi, Widmer, & Zorzi, 2007). Therefore, duplicate sensitive aggregation functions will not return the same result when the data set contains duplicate values but duplicate insensitive functions will. Examples of duplicate sensitive functions include MEDIAN, AVEAGE, SUM, and COUNT. Examples of duplicate insensitive functions include MIN, MAX, and COUNT DISTINCT (Kollios, Byers, Considine, Hadjielefttheriou, & Li, 2005).

Exemplary or summary. Exemplary aggregation only returns one representative value present in the dataset while summary aggregation returns a calculated value from all the data in the set (Madden, Franklin, Hellerstein, & Hong, 2002). In a network where packets tend to be lost during transmission, summary values, for example, AVERAGE, and COUNT, maintain much better accuracy than exemplary aggregates, for example, MIN, MAX, and MEDIAN.

Monotonic aggregates. Aggregates that allow early testing of predicates (such as HAVING in SQL) in the network are monotonic (Madden, Franklin, Hellerstein, & Hong, 2002). The result is the value satisfying the query predicate in the dataset. For example, in a query that requests the MAX temperature reading in the network, as source nodes report their values toward the host node, only the nodes having their value greater than the current MAX will report. This method reduces the traffic without affecting the result.

Partial state requirements. Before the aggregated results reach the sink (in partial state), information may be stored in different formats by aggregate functions. Aggregates such as SUM and COUNT require partial state records that are of the same size as the final aggregate. The AVERAGE function requires a partial state record containing two values (both SUM and COUNT) (Madden, Franklin, Hellerstein, & Hong, 2002). Other aggregates such as MEDIAN and HISTOGRAM require that the entire dataset be returned unless some type of compression or estimation is used.

Data Representation

Each node has limited storage capability to temporarily store received or generated information, including data and parameters about the data, such as type, time, and location sensed, etc. Therefore, the data structure should be concise but rich enough for a node to make decisions about whether to store, discard, compress or transmit the data (Bezenchek, Rafanelli, & Tininini, 1996). It has been pointed out that data structure should be adapted to not only application purpose, but also specificities of devices or node locations (Bezenchek, Rafanelli, & Tininini, 1996). Distributed source coding (Xiong, Liveris, & Cheng, 2004) technique has been proposed to compress data basing on its correlation. Parameters about how data correlates are represented in the data structure.

Performance Evaluation

In essence, in-network aggregation techniques utilize the correlation of data generated by different information sources (sensor units) to realize a trade-off between communication cost and computation cost, and a trade-off between data quality and network lifetime. Therefore, DoA depends on the correlation in data delivery model. Such correlation can be *spatial, temporal,* and *semantic* (Fasolo, Rossi, Widmer, & Zorzi, 2007). Most data models have spatial correlations, in which sensors close by in distance generate more related data. Temporal correlation is an attribute of continuous monitoring model, in which data generated during a small period of time are more related. Semantic correlation is related to information fusion where packets with different contents are the classified under same semantic group (e.g., data generated in the same room, such as humidity, temperature, and brightness) (Fasolo, Rossi, Widmer, & Zorzi, 2007).

Two extreme cases can be used to illustrate the gains of data aggregation: 1. K sources are in a cluster and away from the sink. Meanwhile, sources generate identical results which can be combined into a single packet. In this case, DoA is K. 2. Sources are uncorrelated at all. Sensed data are sent to the sink using certain path. In recent years, models to describe the spatial correlations in terms of joint entropy have been proposed in several papers (Pattem, Krishnamachari, & Govindan, 2004; Pham, Kim, & Moh, 2004; Wang, Pottie, Yao, & Estrin, 2004; Al-khdour, & Baroudi, 2007; Galluccio, Palazzo, & Campbell, 2008; Lu, X., Spear, Levitt, & Wu, 2008). Pattem, Krishnamachari, and Govindan (2004) concluded that for the uncorrelated case, the best routing strategy is to forward packets along shortest paths, whereas for the completely correlated case, the best routing strategy is to aggregate as soon as possible and then forward the result in a single packet to the sink along the shortest path. For cases between the two extreme cases, Pattem, Krishnamachari, and Govindan (2004) suggest that cluster-based solution may be the best choice though no proof was given in the paper.

Usually, performance of an aggregation protocol is evaluated according to metrics of resource efficiency, data quality, reliability, and security in descending importance.

Resource efficiency. Sensor nodes have limited resource supplies, such as power, CPU, bandwidth, memory, etc. Efficient and balanced utilization of those resources leads to energy consumption reduction, which is the one of the main purposes of data aggregation techniques. Energy consumption can be useful or wasteful. The former type includes: transmitting/receiving data, processing query requests, and forwarding query/data to neighboring nodes (Younis, & Fahmy, 2004). The latter type includes: idle state and retransmitting due to packet collision (Younis, & Fahmy, 2004). A number of protocols, especially hierarchical aggregations, have been proposed to reduce useful energy consumption (Heinzelman, Chandrakasan, & Balakrishnan, 2002). The impact of data correlation on energy expenditure has been studied by Zhu, Sundaresan, and Sivakumar (2005). The conclusion obtained was in agreement with Pattem, Krishnamachari, and Govindan (2004) but with quantitative proofs. The study found that Minimum Steiner Tree (MST) has better aggregation effectiveness than Shortest Path Tree (SPT), whereas SPT guarantees low delays. Besides, opportunistic aggregation is compared with systematic aggregation by the *cost ratio* of *correlation unaware* (SPT) tree over *correlation aware* (MST) tree. The cost ratio increases at $O\sqrt{\log N}$ (where N is the number of nodes in the network). The result makes SPT an acceptable aggregation routing structure for a small network.

Major metrics for evaluation include energy consumption of the whole network and network lifetime.

Energy consumption. Energy consumption of a node is mainly composed of communication (transmitting and receiving) and computation cost (sensing and performing aggregation function) typically in a decreasing order in amount (Bai, & Jamalipour, 2008). For a sensor node, the dominating power consumption is transmission cost (Estrin, Sayeed, & Srivastava, 2002). Communication cost can be saved by data aggregation and sleep scheduling. However, aggregation techniques trade off communication cost for computation cost. Setting parameters in aggregation protocols appropriately is critical in balancing both costs and ultimately reaching minimum total consumption. Besides, deploying nodes according to data distribution can help balance overall energy consumption in the network. However, it requires prior knowledge about data distribution model, which is usually not possible in real cases.

Network lifetime. Although many protocols proposed minimize energy consumption by enhancing energy efficiency, such protocols may not prolong network lifetime when some nodes have heavier loads than the others. Network lifetime is defined as the time duration before the sensor network fails to carry out the mission due to insufficient number of "alive" sensor nodes (Akyildiz, Su, Sankarasubramaniam, & Cyirci, 2002). Another popular definition is the time elapsed until the first node (or last node) in the network depletes its energy (dies) (Younis, & Fahmy, 2004). Network lifetime is more than measuring energy efficiency of each node since a node is not rechargeable. It also reflects the overall efficiency of energy distribution across the network (Haenggi, 2003).

Others. Other less used metrics to evaluate energy efficiency include bandwidth, information throughput and total number of active nodes (Boukerche, Cheng, & Linus, 2005). Information throughput usually means the number of packets delivered to the sink. Total number of active nodes indicates the number of alive nodes. Failed nodes are due to insufficient energy to generate packets that meet or exceed a certain threshold value (Boukerche, Cheng, & Linus, 2005). This metric is related to network lifetime in some degree.

Information quality. Decision makers make decisions based on *quality of information* (QoI) that is available to them. The metrics of QoI are different at data collection level, aggregation level and decision making level. To evaluate aggregation effect on information quality, attributes including data *timeliness (freshness)* and *accuracy* are considered (Bisdikian, 2007).

Data timeliness. Timeless describes how timely the data are provided to be useful to applications (Blasch, & Plano, 2005). This attribute is especially important for monitoring applications, which require either continuous real-time monitoring or periodical-based and conditional-based online analytical results about the environment (Guo, Ai, Wang, Cai, & Li, 2009). It is also critical for observer-initiated applications, where timely response is needed. Examples of metrics for timeliness includes the number of transmissions before a packet successfully reaches the sink (Joo, & Shroff, 2008), and *makespan*, the last time slot of the entire aggregation process in which the application receives data from its last child (Yu, Mehrotra, & Venkatasubramanian, 2007).

Data are outdated when arrive at sink mainly because of end-to-end delay and retransmission due to packet loss. Compared with prompt delivery of data to sink, end-to-end delay is significant in hierarchical, especially tree structured network. Except from extra communication time and computation time, routing paths also increase the time spent. Besides, complicated aggregation algorithm or function could worsen the situation by taking long computation time. If it takes long time in each round from spreading query to receiving complete information, throughput is limited in fixed network lifetime.

Solutions to improve freshness of data have been proposed. End-to-end delay may be alleviated by aggregating data as soon as possible and then send to the sink right away, such as cluster approach (Guo, Ai, Wang, Cai, & Li, 2009). Besides, appropriate scheduling technique (Yu, Mehrotra, & Venkatasubramanian, 2007) and balance end-to-end delay using density function (Galluccio, & Palazzo, 2009) can also be used to achieve time efficiency. Packet loss delay may also be minimized by multi-path approach (Joo, & Shroff, 2008).

Data accuracy. *Accuracy* is usually defined as the degree of conformity of a measured or computed quantity to its actual (true) value. Accuracy is related to *precision* that is the degree to which further measurement or calculations show the same or similar results (*ISO/IEC Guide 99-12,* 2007). Resource efficiency, timely delivery of data and data accuracy are conflicting goals. Optimal leverage among them should be depending on specific application (Fasolo, Rossi, Widmer, & Zorzi, 2007; Li, Bandai, & Watanabe, 2010).

Loss of accuracy in data aggregation is mainly due to the reduced size of original data at the sink. With reduced accuracy, the sensed data cannot be completely recovered at the sink (Fasolo, Rossi, Widmer, & Zorzi, 2007). Therefore, data accuracy is closely related to information integrity at the sink. It could happen when the nature of the aggregation function is reductive by generating less representative values, such as AVERAGE. On the other hand, aggregation function without size reduction preserves original information. Fusing different data type or concatenating multiple same type packets into one can reconstruct original data at the sink, such as MEDIAN (Fasolo, Rossi, Widmer, & Zorzi, 2007). Data accuracy can also trade off with data freshness. Solis and Obraczka (2006) classified period synchronization algorithm into three types: *Periodic simple*, *periodic per-hop* and *periodic per-hop adjusted*. In periodic simple aggregation, each node waits for a predetermined amount of time, performs aggregation, and then forwards to the higher level. Since the node may not be able to collect packets from all the child nodes before aggregation, sink only receives partial data in both size reductive or non-reductive case. Accuracy is lost while freshness of data is guaranteed.

There are various ways to measure accuracy. In an energy-accuracy tradeoff algorithm proposed by Boulis, Ganeriwal, and Srivastava (2003), accuracy is estimated from the predefined threshold value, which is equal or greater than actual accuracy, or indirectly from the variance associated with the mean estimate, which is the certainty associated with estimate. Jagyasi, Dey, Merchantand, and Desai (2006) studied the tradeoff between lifetime and accuracy with accuracy calculated as the probability for each sensor to correctly detect an event hypothesis Li, Bandai, and Watanabe (2010) researched on the leverage effects among energy efficiency, data accuracy and timeliness with accuracy calculated as the ratio of number of collected data at sink over number of sensed data at source node.

Recently, new performance metrics considering both data accuracy and aggregation degree were proposed. By introducing the information theory of joint entropy, Pham, Kim, and Moh (2004) proposed a new metric named *Data Aggregation Quality (DAQ).* DAQ is a ratio of total bits after fusion over the joint entropy of all the sensed data to be aggregated. DAQ reflects both aggregation efficiency and the quality of a specific algorithm. However, energy efficiency is not considered. Galluccio, Palazzo, and Campbell (2008) studied the relationship between aggregation and energy consumption as well as the effect of aggregation on information integrity using entropy estimation. Formulas were given to evaluate the effectiveness of aggregation function that can be used to aggressively achieve energy reduction while preserving information integrity.

Reliability. Reliability describes how much confidence can be placed on the data received at the sink. Reliability problem caused by aggregation is mainly due to message lost in transport while traveling hop-by-hop towards the sink. It happens due to the natural lossy characteristics of wireless links (Benson, Roedig, Barroso, & Sreenan, 2007). As aggregated packets carries intense information than normal ones, network reliability is important for maintaining data accuracy and timeliness. Benson, Roedig, Barroso, & Sreenan, 2007) proposed a reliability control mechanism, in which end-to-end reliability is calculated from the data transport reliability described by sensor readings per unit time reaching the sink and also by variance over expected value.

Security. The metric of security is important to applications that involve sensitive or private data detection. Since data are transported wirelessly between nodes, they are susceptible to interception and eavesdropping (Bista, Jo, & Chang, 2009). Many security protocols for aggregated WSN were proposed to solve *external security*, referring to protecting sensed data from outsiders, such as adversaries (Bista, Jo, & Chang, 2009). Besides, Blass, Wilke, & Zitterbart, 2006) stated that security in aggregated sensor network should not only imply confidentiality of transported data, but also the authenticity or originality of participating (aggregating) nodes. Examples of metrics correctness, authentication (Blass, Wilke, & Zitterbart, 2006) and communication overheard. Communication overheard (Bista, Jo, & Chang, 2009) is measured against number of messages generated and energy consumption to evaluate the security of the aggregate protocol.

Literature Review

Tree-based Protocols

Tree-based aggregation is also referred to as hierarchical or classical routing approach. The structure of the network is based on a tree rooted at the sink (Fasolo, Rossi, Widmer, & Zorzi, 2007). Data are aggregated while flowing from the source to the sink along the tree branches at selected nodes. Such a node is selected based on various factors depending on the application, such as its hierarchical level, its position (Solis, & Obraczka, 2005), resource (Erramilli, Malta, & Bestavros, 2004), data type (Ding, Cheng, & Xue, 2003), and aggregation cost (Luo, Luo, & Liu, 2005; Galluccio, Palazzo, & Campbell, 2008).

Aggregation schemes of the tree-based approach includes *opportunistic*, *greedy-incremental* and *optimal aggregation* (Intanagonwiwat, Estrin, Govindan, & Heidemann, 2002; Krishnamachari, Estrin, & Wicker, 2002). In opportuninistic aggregation, when similar data happens to meet at a branching node in the tree, data compression is performed (Intanagonwiwat, Estrin, Govindan, & Heidemann, 2002). Opportunistic aggregation is not optimal regarding energy saving because data may be aggregated hops away from the source. An optimal solution will be aggregating similar data as soon as possible after they are collected by sensors. Greedy incremental tree (GIT) (Intanagonwiwat, Estrin, Govindan, & Heidemann, 2002) differs from opportunistic aggregation in path establishment and maintenance. A shortest path is connected for the first source node to the sink. Incrementally, other subsequent source nodes are connected based on their distances to the existing tree. The tree-based approach is suitable for designing optimal aggregation functions and performing efficient energy management. However, it is not efficient in the case of dynamic topology and link failure where expensive reorganization is needed. Besides, it is also unreliable due to the sensitivity to any failure of intermediate nodes, in which case, data from all child nodes are lost (Fasolo, Rossi, Widmer, & Zorzi, 2007).

Multi-path Protocols

Muti-path approach (Nath, Gibbons, Seshan, & Anderson, 2004; Manjhi, Nath, & Gibbons, 2005; Chen, & Zhang, 2006) is relatively new. These protocols are proposed to increase the reliability of the network. They are suitable for applications vulnerable to packet loss due to mobility or channel impairments. Multiple paths have been established between the source node and the sink node. The multi-path approach has been classified into two modes: the primary/back mode (Huang, & Jan, 2004) and the replication mode (Deb, Bhatnagar, & Nath, 2003). The primary/backup mode uses backup paths to transmit data when the primary path is not available. The replication mode takes advantage of the broadcasting feature of sensor nodes. Data is transmitted through all the paths simultaneously. It allows propagating duplicate information. Therefore, trade-off exists between network robustness and overhead due to duplicated data. This mode is easier to implement, but may suffer from higher energy consumption and traffic congestion (Dulman, Nieberg, Wu, & Havinga, 2003). Both modes do not efficiently combine load balancing and energy awareness in sensor networks (Yang, Lin, Xiong, &

Xu, 2009). Algorithms that route data through a path whose nodes have the most residual energy are extensively studied (Shah, & Rabaey, 2002; Chang, & Tassiulas, 2004). However, it could be expensive too to compute and pick the optimal path. There is a tradeoff between minimizing total energy consumption and residual energy of the network (Li, Aslam, & Rus, 2001).

Multi-path approach can be implemented through ring structure or multiple spanning trees such as Directed Acyclic Graph (DAG) (Motegi, Yoshihara, & Horiuchi, 2006).

Ant Colony Optimization Algorithm

The Ant Colony Optimization (ACO) algorithm is a metaheursitic initially proposed by Marco Dorigo in his PhD dissertation in 1992. “The original idea comes from observing the exploitation of food resources among ants, in which ants’ individually limited cognitive abilities have collectively been able to find the shortest path between a food source and the nest” (“Ant colony optimization algorithms”, 2011).

It is firstly used to solve traveling salesman problem (TSP). Because of the characteristics of distributing computing, self-organization and positive feedback, ACO has been used in prior works for routing in sensor networks (Das, Singh, Gosavi, & Pujar, 2003; Yang, Lin, Xiong, & Xu, 2009). In 2006, Misra and Mandal firstly introduced Ant-aggregation algorithm using ACO for optimal data aggregation in wireless sensor networks. “Node Potential” is the heuristic used to evaluate the potential of next hop selection based on three factors: the candidate’s distance to the sink node, its distance to the nearest aggregation node and its data correlation with the current node. In this algorithm, random searching for the destination (sink node) is needed in early iterations. In addition, dead lock occurs when ants travel in a cycle.

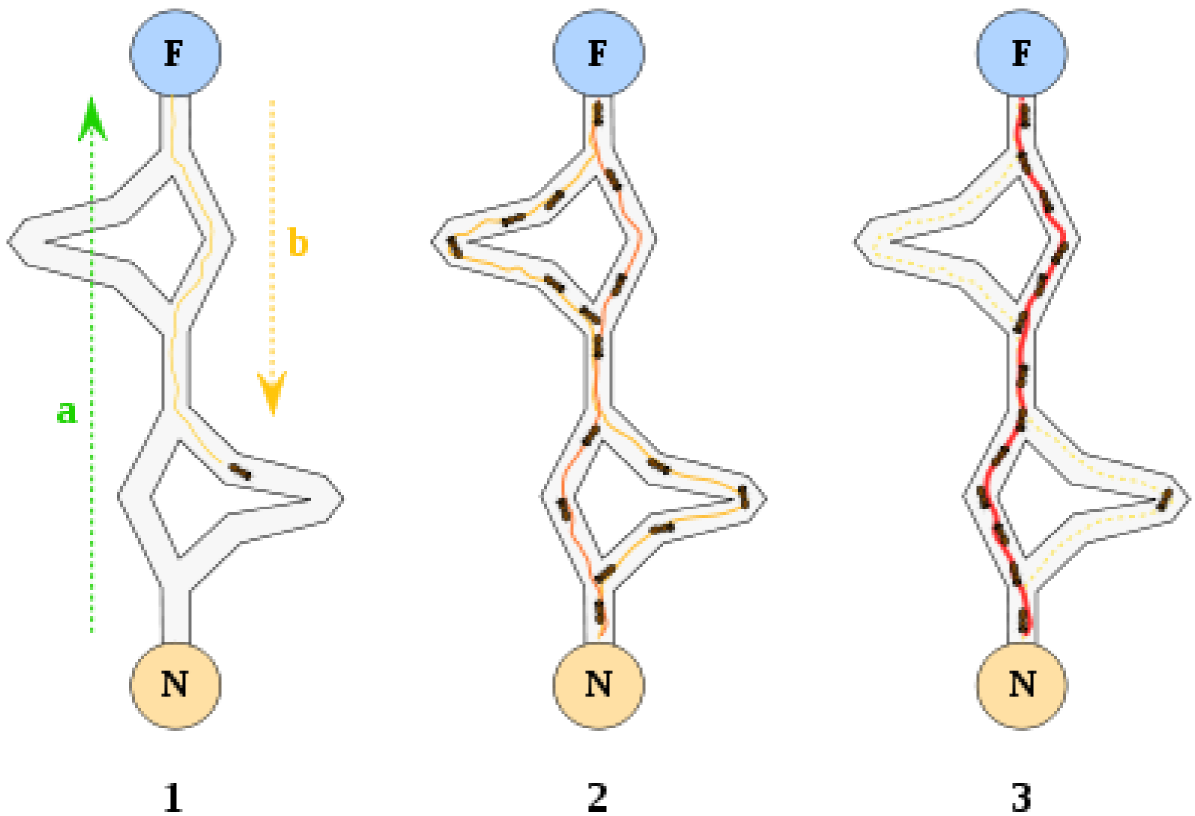

Figure 1. The exploitation of food resources among ants.
("Ant colony optimization algorithms," 2011)
("The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b). Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route. Ants take the shortest route, long portions of other ways lose their trail pheromones" ("Ant colony optimization algorithms," 2011))

Afterwards, many improved algorithms have been proposed. Those algorithms use a simpler heuristic by only considering the distance to the sink node. Liao, Kao, and Fan introduced an algorithm in 2007, which makes it easier to seek aggregation node. Chen, Guo, Yang, and Zhao (2006) proposed an approach using search ants at the beginning for the searching of destination. Wang and Luo (2010) proposed an algorithm composed of path construction, path maintenance, and aggregation schemes including synchronization scheme, loop-free scheme, and avoiding collision scheme.

There is a problem ignored by the algorithms above. Although ACO aggregation algorithms converge to a route very close to the optimum route, most of them only use a single path to transfer data until an active node in the path runs out of battery. Then the path construction and data delivery cycle starts again. Although route discovery overhead can be reduced, those algorithms do not taken into consideration limitations of WSNs, especially energy limit of sensor nodes and number of agents required to establish the routing (Yang, Lin, Xiong, & Xu, 2009). Repeatedly using the same optimal path exhausts the relaying nodes' energy quickly. Relatively frequent efforts to maintain the network and to explore new paths are needed. Therefore, this approach is not energy efficient and results in shorter sensor nodes' lifetime and consequently network lifetime. Algorithms that separate path establishment and data delivery processes suffer from this problem.

Another application of Ant Colony Optimization in wireless sensor networks is in multi-path routing. Multi-path routing algorithms based on ACO have been proposed in recent years. Xia, Wu, and Ni (2009) incorporated three new rules in the algorithm to solve the problems of local convergence, local optimization, and multi-path for transferring data, respectively. Xia and Wu (2009) proposed an energy-aware multi-path routing algorithm that considers the available power of nodes and the energy consumption of each path as the reliance of routing selection. Yang, Xiong, and Xu (2009) propose a load balancing scheme to distribute the traffic over multi-paths discovered. In each paper, network lifetime is proved to be longer compared with traditional ACO methods. However, these routing algorithms do not aggregate data in the routing process.

Data aggregation approach improves energy efficiency in wireless sensor networks by eliminating redundant packets, reducing end-to-end delay and network traffic. This research studies the effect of combining data aggregation technique and multi-path ACO algorithm with different heuristics on network lifetime and end-to-end delay.

Algorithm overview. In ACO algorithms, a colony of artificial ants is used to construct solutions guided by the pheromone trails and heuristic information. The original idea of ACO comes from observing the exploitation of food resources among ants. Ants explore the area surrounding their nest initially in a random manner. As soon as an ant finds a source of food (source node), it evaluates the quantity and quality of the food and carries some of it to the nest (sink node). During the back tracking, the ant deposits a pheromone trail on the ground. The quality of deposited pheromone, which may depend on the quantity and quality of the food, will guide other ants to the food source (Liao, Kao, & Fan, 2007). The pheromone trails are simulated via a parameterized probabilistic model. The pheromone model consists of a set of parameters. The basic component of the ACO algorithm is “a constructive heuristic that is used for probabilistically constructing solutions using the pheromone parameters” (Misra & Mandal, 2006).

In general, the ACO approach attempts to find the optimal routing by iterating the following two steps: (1) Solutions are constructed using a node selection model based on a predetermined heuristic and the pheromone model, a parameterized probability distribution over the solution space; (2) The solutions that were constructed in earlier iterations are used to modify the pheromone values in a way that is deemed to bias the search toward high quality solutions (Misra, & Mandal, 2006).

The algorithm runs in two passes: forward and backward. In the forward pass, the route is constructed by a group of ants, each of which starts from a unique source node. In the first iteration, an ant searches a route to the destination randomly. Later, an ant searches the nearest point of the previously discovered route. This could take many iterations before the ant can find a correct path with a reasonable length. A solution is flooding the sink node ID from the sink to all the sensor nodes in the network before any ant starts (Wang, & Luo, 2010). The points where multiple ants join are aggregation nodes. In the backward pass every ant starts from sink node and travels back to the corresponding source node by following the path discovered in the forward pass. Pheromone is deposited hop by hop during the traversal. Nodes of the discovered path are given weights as a result of node selection depending on the node potential which indicates heuristics for reaching the destination. Pheromone trails are the heuristics to communicate with other ants of the route discovered. The trail followed by ants most often gets more and more pheromone and eventually converges to the optimal route. Pheromone in non-optimal route gets evaporated with time. The aggregation points on the optimal tree identify data aggregation. The indicator in each data aggregation point gives estimate of the number of paths aggregating in the point.

My Research Work on Ant Colony Optimization Algorithm

This thesis project introduced multi-path approach and in-network data aggregation scheme into the Ant Colony Optimization algorithm in order to extend network lifetime. In this chapter, three algorithms, ”SinkDistComb”, “ResidualEnergy”, and “SinkAggreDist,” are proposed and compared with the conventional algorithms. The path discovery procedure and next node selection used in the Ant Colony Optimization algorithm are explained. In the end, the measurement of network lifetime is defined.

Algorithms

This thesis project compared five ACO algorithms that utilize different node selection rules, aggregation schemes, and heuristics. The algorithms guarantee packets delivery. The routes discovered are very close to the optimum one.

1. “SinkDistNoAggre”
2. “SinkDistLead”
3. “SinkDistComb”
4. “ResidualEnergy”
5. “SinkAggreDist”

These algorithms follow the major path discovery procedure of ACO. The procedure is illustrated by figure 2.

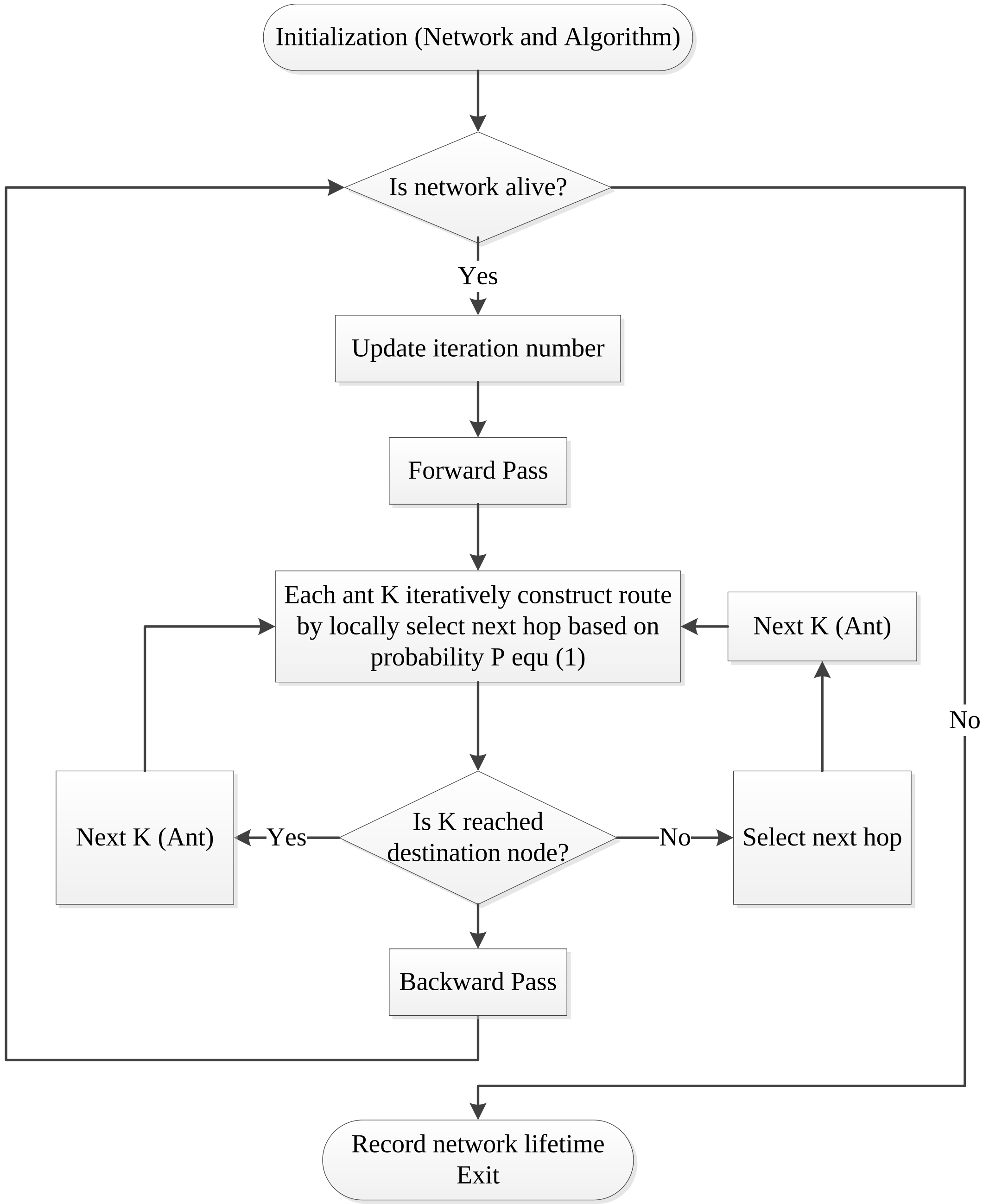

Figure 2. Ant Colony Optimization algorithm

"SinkDistNoAggre" algorithm. The "SinkDistNoAggre" algorithm is the most similar one to the conventional ACO algorithm. When selecting the next hop, all the neighborhood nodes of the sending node are considered as candidates. The selection is based on the candidate node's probability value. The probability calculation equation is set by the node selection rule. The heuristic used in the rule is the candidate node's distance or hop-count to the sink node. The node with the highest probability is selected. This algorithm does not perform data aggregation during routing process.

"SinkDistLead" algorithm. In the "SinkDistLead" algorithm, the heuristic is the candidate node's distance to the sink node. The node with the highest probability is selected. This algorithm performs data aggregation.

"SinkDistComb" algorithm. In the "SinkDistComb" algorithm, the heuristic is the candidate node's distance to the sink node. A group of candidates with the highest probability is found using the probability equation and cached. In every iteration, one candidate node is randomly selected and then removed from the cache. If the cache is empty, find the group of best candidates again. This algorithm performs data aggregation.

"ResidualEnergy" algorithm. The "ResidualEnergy" algorithm is an energy-aware multi-path algorithm including data aggregation. The heuristic includes: 1) the node's distance to the sink node and 2) the node's remaining energy. The node with the highest probability is selected. This algorithm performs data aggregation.

"SinkAggreDist" algorithm. The "SinkAggreDist" algorithm encourages early aggregation. The heuristic includes: 1) the node's distance to the sink node and 2) the distance from the candidate node to the nearest aggregation node in the last iteration. A group of candidates with the highest probability is found using the probability equation and cached. In every iteration, one candidate node is randomly selected and then removed from the cache. If the cache is empty, find the group of best candidates again. This algorithm performs data aggregation.

This paragraph explains how each sensor node finds the nearest aggregation node. Initially, each sensor node does not have the nearest aggregation node. At the end of each iteration, each aggregating node in the last forward pass floods its ID to all the nodes that have greater distance to the sink node. Only the nearest aggregating node is remembered by each sensor node. Therefore, the nearest aggregation node remembered is updated every iteration. Besides, the aggregation node must be closer to the sink node than the sensor node itself. If a sensor node is an aggregation node itself, it remembers its own ID.

Path Discovery Procedure

The procedure is mainly composed of forward and backward passes. In the forward pass, an ant tries to explore a new path based on the heuristic rule and the pheromone amount on the edges. Backtracking is used in the forward pass when an ant finds a dead end or is running into a loop. In the backward pass, the ant updates the pheromone amount on the path constructed in the forward pass. Other important components in the algorithms include data-aggregation, loop control, and network maintenance.

In WSN, each node has a unique identity. Every node is able to calculate and remember its current heuristic value. Initially, the sink node floods its identity to all the nodes in the network. After a node receives the packet, it computes its hop-count to the sink node and correspondingly its initial heuristic value. The heuristic is updated whenever the value used is changed, such as the hop-count to the sink node, the node's residual energy, or the distance to the nearest aggregation node.

Forward pass. Each ant is assigned a source node. After that, an ant starts from the source node and moves towards the sink node using ad-hoc routing. The forward pass ends only if all the ants have arrived at the sink node. Single ant-based solution construction uses following steps:

- If the node has been visited in the same iteration, follow a previous ant's path; or
- Use a node selection rule; or
- If all the neighbors have been visited, use the shortest path; or
- If no neighbor nodes, backtrack to the previous node ; or
- If no neighbor nodes and the previous node is dead, record the network lifetime and exit the program.
- Transmit the packet.

The current node sends the packet. The selected node receives the packet. Both nodes update the residual energy after transmission. If the current node does not have enough energy to send, this transmission fails. The network is maintained afterwards. Transmission failure is mostly prevented by doing a receiving and sending energy check in the node selection step.

Backward pass. Ants start from the sink node and move towards their source nodes. The ants follow the paths discovered in the forward pass. Before an ant arrives at its source node, the algorithm repeats:

- Retrieve the previous node in the path solution.
- Transmit the packet.

- If transmission fails, maintain the network and terminate this ant.
- Encourage or discourage the node selection in the forward pass by depositing or evaporating the pheromone on the edge.

Data aggregation in forward pass. Each sensor node maintains two queues to store packets: a receiving queue and a sending queue. The packet sending process includes:

- Remove all the packets from the receiving queue.
- For ”SinkDistNoAggre”, push all the packets into the sending queue.
- For other aggregation algorithms, use the predefined function to aggregate all the received packets into one packet and push it into the sending queue.

Among all the ants arrived at this node, select the earliest ant as the aggregating ant. The aggregating ant will finish the rest of the routing construction in this iteration. All the later arrived ants become aggregated ants. They remember the aggregating ant. Each aggregated ant shares its path with the aggregating ant. The aggregating ant updates its subsequent hops with all the aggregated ants.

Loop control and failure handling. “Loop” is defined as the situation that an ant revisits an already-visited node in the same forward pass. Since each ant remembers the path, it can avoid running into a loop by comparing the candidate node’s ID with the visited nodes’ IDs.

An ant is considered failing its task in a iteration, if all the neighborhood nodes of the current node have been visited. In that case, the ant uses the shortest path to deliver the packet to the sink node. The node’s previous visiting history is not considered when choosing the next node. A path resulting in “failure” is discouraged.

Network maintenance. When a node does not have sufficient energy to send or receive (the "dead node"), it is removed from the neighbor list of its neighborhood nodes. Nodes with more hop-count than the "dead node" recalculate their hop-count and heuristic value. If the "dead node" is a source node, find the node with the maximum energy in the network as the new source. Afterwards, update the source node of the ant. If the "dead node" is the sink node, recharge the node with more energy. Sink node is different from other nodes because it needs to perform more frequent transmission and computation for the purpose of application. Therefore, it is assumed that the sink node has plenty energy to last until the network dies.

Next Node Selection

To support next node selection, rules are established and followed in the forward pass. These rules check the candidate node's probability calculated from heuristic and pheromone values. The heuristic is updated whenever the value is changed. The pheromone is updated in the backward pass according to the rules set.

Node selection rules. Two rules are used for next node selection: "LeadingExploration" and "CombinedRule". The "SinkDistNoAggre," "SinkDistLead," and "ResidualEnergy" algorithms use "LeadingExploration" because the first found best candidate node needs to be selected. The "SinkDistComb" and "SinkAggreDist" algorithms use "CombinedRule" so that multiple paths can be established.

"LeadingExploration." Among all the neighborhood nodes, select the first node with the highest probability, even if there are multiple nodes with the same probability. This method is deterministic. In every iteration, an ant always discovers the same path to the sink node until one of the intermediate nodes dies. If the same network topology is tested repeatedly, the total energy cost and network lifetime are the same.

"CombinedRule." Node selection is divided into sessions. Each session includes one or more iterations. A node discovered from the current or a previous iteration is used. Similar to "LeadingExploration," the probability of each neighborhood node is calculated. A group of nodes with highest probability is stored in a cache. In each iteration, one node is randomly selected and removed from the cache. When the cache is empty, the probability calculation of all the alive neighborhood nodes is repeated.

Probability calculation. When a node is ready to send a packet, it calculates the probability of all the neighbors using the equation below.

$$p_k(i,j) = \frac{\tau(i,j) \times \eta(i,j)^{\beta}}{\sum_{u \in N_i} \tau(i,u) \times \eta(i,u)^{\beta}} \qquad (1)$$

In equation (1), an ant k having data packet in node i chooses to move to node j until the sink node, where τ is the pheromone, η is the heuristic, Ni is the set of neighbors of node i, and β is a parameter which determines the relative importance of pheromone versus distance (β>0). Value η is calculated using equation (2). Multiple factors can be used and each one is weighted.

$$\eta\ (i,j) = \sum_{\substack{sum\ of\ weight\ k=1 \\ 0 \leq weight\ k \leq 1}} Cost\ k * \ weight\ k \qquad (2)$$

$Cost_{\ dist.\ to\ sink}$ – inverse of the distance between node j and the sink plus one.

$Cost_{\ residual\ energy}$ – the left energy of the candidate node.

$Cost_{\ dist.\ to\ aggre.\ node}$ – inverse of the distance between node j and the aggregation node plus one.

Pheromone update rules. The pheromone value is associated with the link (edge) between two nodes. Each edge has a pheromone value, which is initially all the same. The value is updated in each iteration in order to bias the node selection process in the next iteration. The value is updated twice in each iteration.

Evaporation on all edges. After all the ants finish the forward passing and before they are going backward, the pheromone values on all the edges in the network evaporate at rate ρ. The value is consistently reduced. Equation (3) shows how the evaporated pheromone value is calculated.

$$\tau_{ij} = (1 - \rho) \times \tau_{ij} \quad (3)$$

Deposit pheromone during backward pass. In the backward pass, each ant deposits or reduces the pheromone value on its own solution path. This step is different from the conventional ACO algorithm, in which pheromone is always deposited using the same rate. Encouraging or discouraging a node choice in the forward pass depends on the comparison of performance in the forward pass with the one of the best iteration found so far. The new pheromone is calculated using equation (4). Equations (5) and (6) are used to support equation (4).

$$\tau_{ij} = (\tau_{ij} + \rho\Delta\, \tau_{ij}) \times e0 \quad (4)$$

$$\Delta\, \tau_{ij} = [\zeta + (h_i - h_j)] \times \Delta\omega_j \quad (5)$$

$$\Delta\omega_j = \sum_{j \in R_j} {H_{ij}}^{-1} + {h_j}^{-1} \quad (6)$$

In equation (4), ρ is the pheromone decay parameter, τ_{ij} is the pheromone value on the edge between nodes *i* and *j,* and e0 is the encouraging or discouraging rate derived from the forward pass. A path resulting in less energy consumption and smaller total hop-count is preferred. The best iteration is one with the least energy consumption and hop-count among all previous iterations. It is used as a control to calculate the e0 in the current iteration. If the forward pass is a failed path exploration or used more hop-count and energy consumption than the best iteration, the path is discouraged. Very small amount of pheromone is deposited on the edge to differentiate from those links not been visited, and e0 is set to a predetermined "punishRate," which is a relatively low rate between 0 and 1. If the forward pass found a path using the same hop-count and energy consumption as the best iteration, e0 is set to a relatively higher rate between 0 and 1--the "encourageRate." If the forward pass found a path with the same hop-count but less energy consumption than the best iteration, e0 = 1.5 × encourageRate. If the forward pass found a path using less hop-count and energy consumption than the best iteration, e0 = hop-count difference × encourageRate.

In equation (5), ζ is a positive number, *hi* is the hop-count between node *i* and the sink, and *hj* is the hop-count between node *j* and the sink. If the value of (*hi* - *hj*) is greater than zero, it can be concluded that node *j* is closer to the sink node than node *i*. Therefore, the algorithm rewards the path from node *i* to node *j* by depositing more pheromone. If the value equals to zero, it means that both nodes *i* and *j* have the same hop-count to the sink, then the algorithm lays little pheromone on the path. If the value is less than zero, the algorithm does not lay pheromone on this path. In equation (6), *Rj* is the set of different ants or sources through node *j*, and $\sum_{j \in R_j} {H_{ij}}^{-1}$ is the total hop-counts of these sources before vising node *j*. Therefore, $\Delta\omega_j$ is the total hop-counts of some sources to the sink through node *j*. The less the total hop accounts, the larger amount of pheromone is added on the path from node *i* to node *j*, as shown in equation (5). This means that more ants are encouraged to follow this path. For an aggregation node, it updates the pheromone levels of its all neighbors by equation (4) when an ant moves to it. If a node does not have ants visit it within a limited time, its pheromone is evaporated according to equation (3).

Network Lifetime

In the five algorithms, the network lifetime is measured using the number of iterations completed up to the moment when any one ant cannot find the next node to send the packet. It indicates all the neighborhood nodes are out of power. When this situation is detected, the algorithm terminates and the performance metric results are recorded.

Simulation and Result

Simulation

The ACO aggregation algorithms were simulated in a custom program tool written in C#. Five algorithms: “SinkDistNoAggre”, “SinkDistLead”, “SinkDistComb”, “ResidualEnergy”, and “SinkAggreDist” were evaluated to study the network lifetime and hop-count delay. The input to each algorithm is a network composed of sensor nodes and a destination node. Each node has a unique ID number. All the nodes have the knowledge of the neighborhood nodes obtained from the random topology. The output of each algorithm is performance metric results. Logging files are maintained for debugging purpose.

A controlling class manages the five algorithms throughout a predefined number of rounds. In each round, the algorithms only apply on one random network topology. The algorithms run independently. The input topology to each of them is exactly the same. After all rounds are finished, the controlling class calculates the average performance results and the program terminates.

Network parameters. The input network is a group of sensor nodes randomly distributed in a 50m × 50m square area. The network density is the ratio of the number of nodes to the area 50 × 50. It represents the percentage of nodes in the confined area. A set of existing nodes are randomly selected as source nodes. The source density is the ratio of the number of source nodes to the total number of nodes. Each node’s position in the network is represented using coordinate (x, y). The node's ID is calculated form the equation: x × 10 + y. The sink node is placed at the top left edge. The routing paths are limited to one dimension of the Cartesian coordinate system to prevent long hop-count delays across multiple dimensions. All the nodes have the same transmission radius. The distance between two nodes is calculated using the Euclidean distance formula. If the distance is less than or equal to the transmission radius, the two nodes are considered neighbors. Table 1 shows the network configuration.

Table 1

Network Parameters

Parameter	Description	Setting
iMaxX	Network X-axis	50
iMaxY	Network Y-axis	50
Sink Node	Destination Node	(0, 0)
iNetworkDensity	Total Nodes/iMaxX × iMaxY	0.3, 0.5, 0.7
iSourceDensity	Source Nodes/ Total Nodes	0.004 – 0.01
iTransmissionRadius	Transmission Range	5, 6, 7, 10, 15, 20
iTransmissionRadius/iMaxX	Relative transmission range	0.1, 0.12, 0.14, 0.2, 0.3, 0.4

ACO algorithm parameters. Table 2 is a list of ACO algorithm parameters. Parameter ζ, η, ρ, e0, encourRate, punishRate, α, β are used in equations (1) to (4) in Chapter 3. The number of rounds is set to 30 in order to minimize the effect of randomness of network topology. In the "ResidualEnergy" algorithm, the heuristic is composed of two factors – the node's distance to the sink node and the node's residual energy. The two factors have different weights and the sum of the weights is 1. Parameter "resEnergyEta" represents the weight of the distance factor. Similarly, in "SinkAggreDist" algorithm, the heuristic is composed of two factors – the node's distance to the sink node and the distance to the nearest aggregation node. Parameter "sinkAggreEta represents the weight of the distance factor.

Table 2

Algorithm Parameters

Parameter	Description	Setting
ζ (Xi)	A positive value	1
η (Eta)	1/distance to sink + 1	-1
ρ (Rho)	Pheromone evaporation rate	0.1
e0	Encourage or Discourage rate	0
encourRate	Encourage rate	0.9
punishRate	Discourage rate	0.1
α (Alpha)	Weight of pheromone	0.5
β (Beta)	Weight of heuristic	0.5
τ0	Initial pheromone	0.35
round	Total times of repetition	30
resEnergyEta	Weight of "distance to sink" in "ResidualEnergy" algorithm	0.5
sinkAggreEta	Weight of "distance to sink" in "SinkAggreDist" algorithm	0.5

Energy model. Every node initially has the same amount of energy. After transmission, certain amount of energy is consumed. The energy model by Heinzelman, Chandrakasan, and Balakrishnan (2000) is used to estimate energy consumption. The energy consumption for sending a packet is determined by a cost function: $Esend = Etrans \times s + Eamp \times d^2$, where *Esend* is the energy cost of sending a bit, *s* is the packet size, *Eamp* is the energy consumed in the amplifier, and *d* is the Euclidean distance of message transmission. The energy consumption for receiving a message is determined by a cost function: $Ereceive = Erec \times r$, where *Etrans* is the energy cost of receiving a bit, and *r* is the packet size.

Table 3

Energy Model Parameters

Parameter	Description	Setting
E*trans*	Transmitter electronics	50 *nJ/bit*
E*amp*	Transmit amplifier	0.1 *nJ/bit/ m2*
E*receive*	Receiver electronics	50 *nJ/bit*
iMaxEnergy	Initial energy of sensor nodes	0.0001 *J*
CP*size*	Control packet size	1*byte*
DP*size*	Data packet size	4*bytes*

Performance metrics. In each round, network lifetime and average delay in hop-count are calculated. As explained in Chapter 3, the network lifetime is defined as the number of iterations completed when any one ant fails to find the next node due to energy outrage of all the neighborhood nodes. Long network lifetime is desired. There is no upper bound of the iteration number, so the Wireless Sensor Network is considered alive as long as the definition of lifetime is satisfied.

Delay is the elapsed time between the moment when a package is sent by the source node and the moment it arrives at the sink node. Short delay is desired so that the sink node receives timely delivered information. Assume the time spent in a hop-to-hop transmission is all the same, delay is measured using hop-counts. The delay between one source node and the sink node is the hop-count in one path solution. The average delay in one round is calculated using the equation below:

$$\frac{\sum_{All\ iterations} TotalHopCountInOneIteration\ /\ TotalSourceNodes}{Iteration\ Count}$$

Simulation Result

The five algorithms are compared with each other in aspects of network lifetime and hop-count delay. For each aspect, effect of transmission range, network density, and source node ratio on the performance of algorithms are discussed.

Network lifetime. Figures 3, 4, and 5 show that “SinkDistComb” and “ResidualEnergy” yield longer network lifetime than “SinkAggreDist” and “SinkDistLead”. The algorithm “SinkDistNoAggre” has significant shorter network lifetime than the algorithms with aggregation. Three main conclusions can be drawn from the simulation results. First, ACO algorithm generates much longer network lifetime when combined with in-network data aggregation scheme. Second, if the same heuristic is used, the multi-path ACO algorithm using “combinedRule” as node selection rule generates longer network lifetime than the single-path algorithm. Third, “SinkDistComb” and “ResidualEnergy” are superior algorithms when network lifetime is considered as the major performance metric.

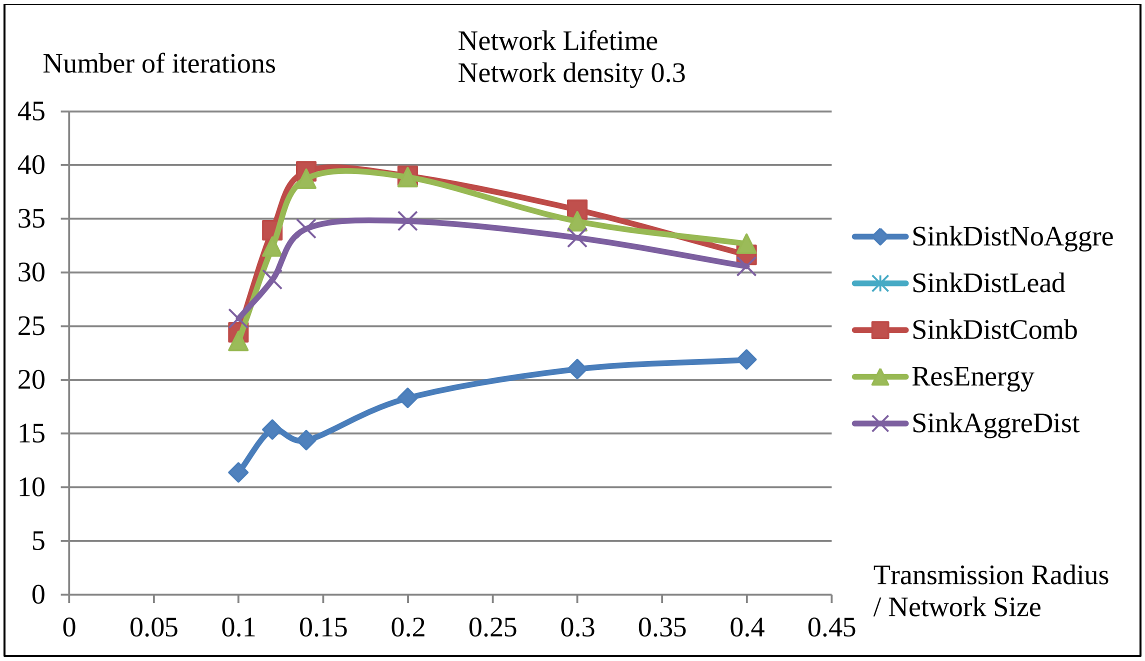

Figure 3. Network lifetime of the five algorithms with network density 0.3, source node ratio 0.01, and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

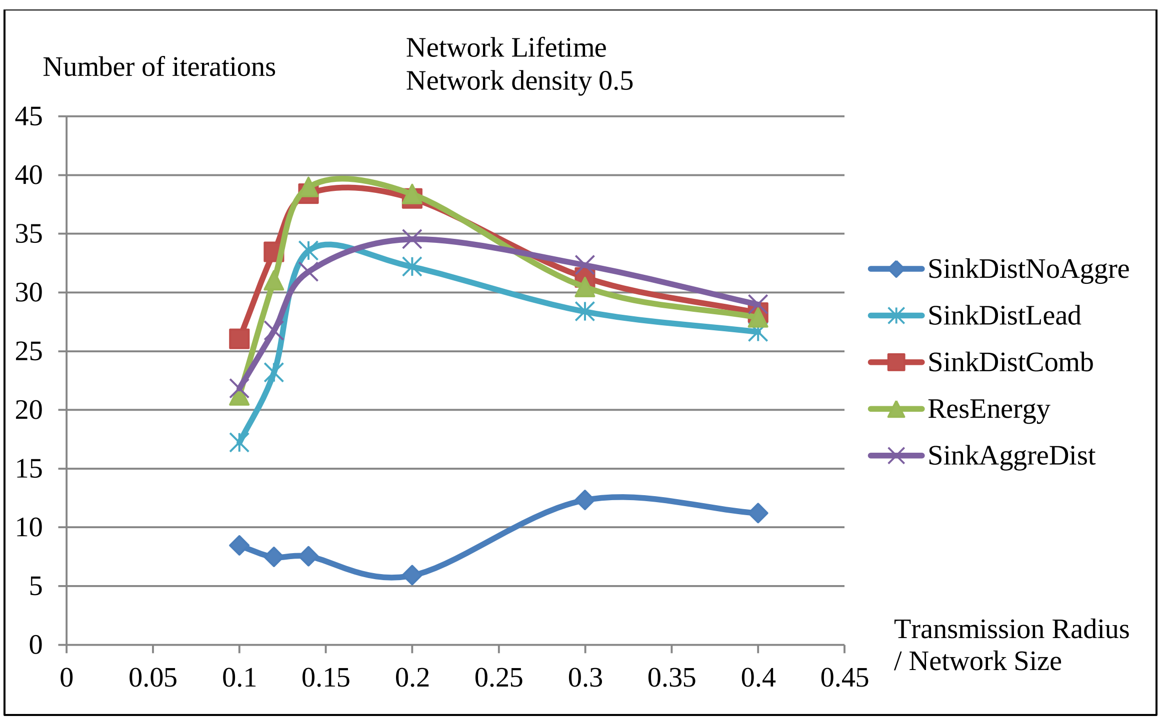

Figure 4. Network lifetime of the five algorithms with network density 0.5, source node ratio 0.01, and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

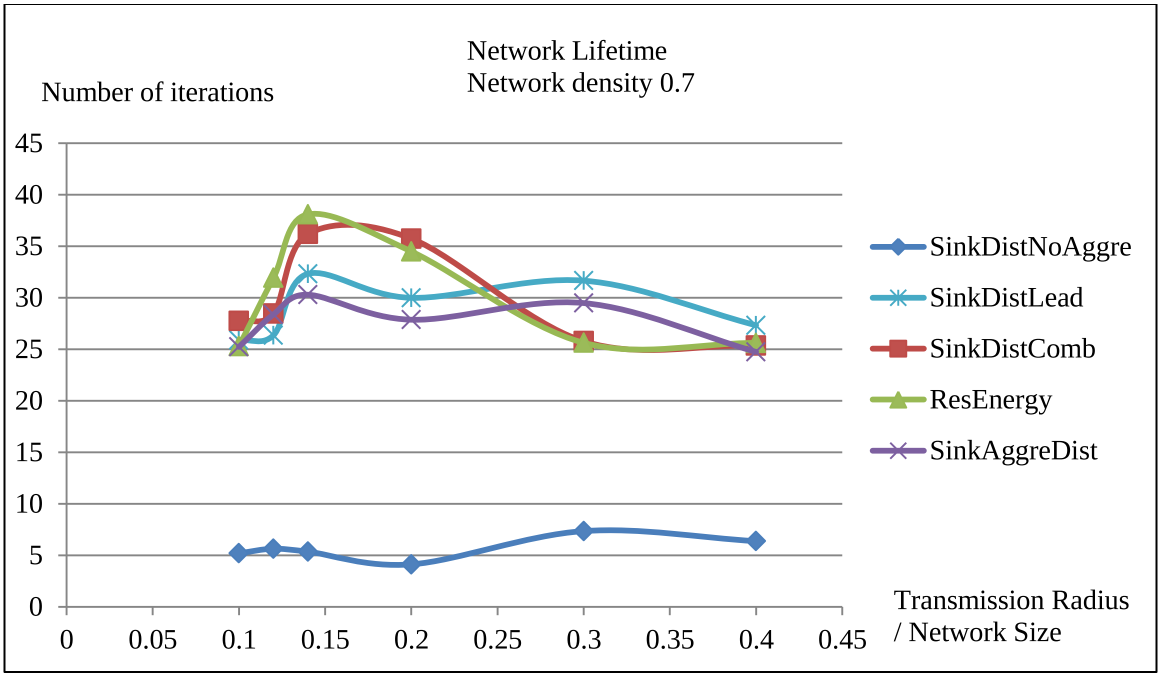

Figure 5. Network lifetime of the five algorithms with network density 0.7, source node ratio 0.01, and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

Transmission range. Figures 3 to 5 show the effect of transmission range on the algorithms “SinkDistComb”, “ResidualEnergy”, and “SinkAggreDist”. The highest network lifetime is achieved at the ratio of transmission radius / network size 0.14. For a 50 × 50 network, the transmission radius is 7. The network lifetime increases quickly when the ratio is raised from 0.1 to 0.14, starts declines around 0.15 and becomes stable around 0.4.

The trend of the curve can be explained by the relationship between transmission range, energy consumption distribution, and early aggregation. The longer transmission range, the greater number of neighborhood nodes a node can have. This allows a sending node to have more candidates when choosing the next relaying node. As a result, it is likely that the sending node has better candidate nodes, such as closer position to the sink node or greater residual energy, compared with having less neighborhood nodes. This explains the quick increase of the network lifetime when the ratio grows from 0.1 to 0.14. However, it is not always true that longer transmission range leads to longer network lifetime. The network lifetime reaches its maximum value at certain transmission range because the energy consumption is most evenly distributed at this range due to early aggregation (Figure 6). Each node in the network spends a similar amount of energy when the algorithm is running. As the transmission radius is longer, aggregations become more aggressive, which means fewer aggregations are needed. Because of larger and overlapping broadcast area of sending nodes, an aggregation node aggregates more packages from different source nodes (Figure 7). This results in the accelerated energy consumption therefore the quick depletion of node's power. More frequent network maintenance is needed. The consequence is the decrease of the network lifetime.

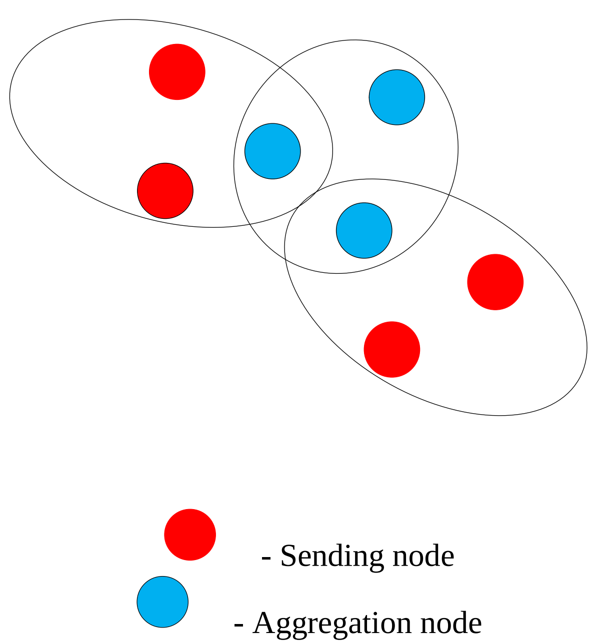

Figure 6. Aggregation with transmission radius 1
(Each aggregation node aggregates 2 packets)

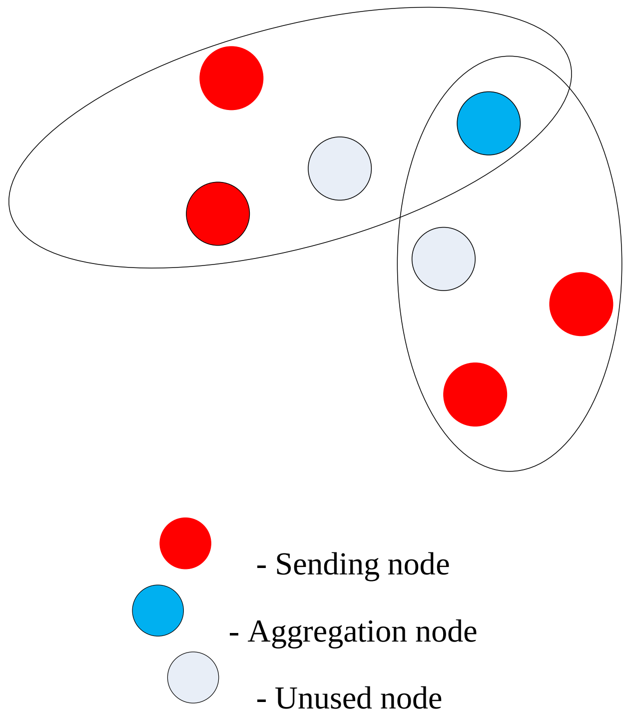

Figure 7. Aggregation with transmission radius 2
(Each aggregation node aggregates 4 packets)

Network density. The effect of network density on the network lifetime for the algorithms "SinkDistComb", "ResidualEnergy", and "SinkAggreDist" is studied in two cases: with same ratio of source nodes and with same number of source nodes.

With same ratio of source nodes. Figures 8, 9 and 10 illustrate the influence of network density on network lifetime with fixed source node ratio.

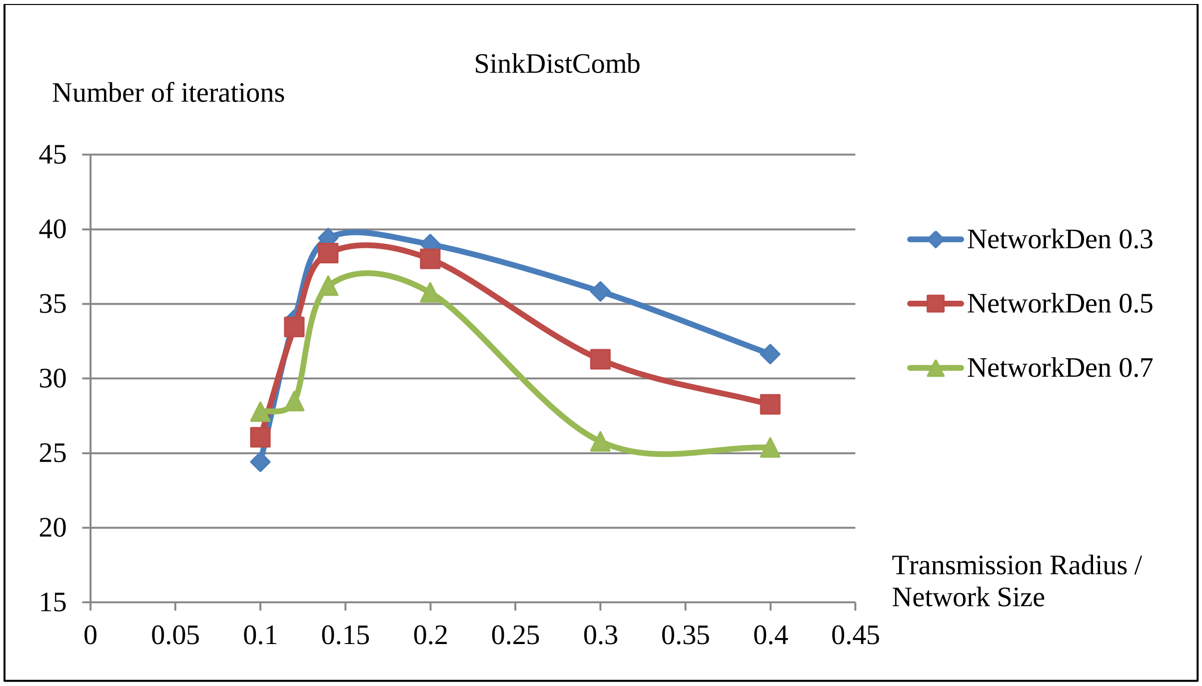

Figure 8. Network lifetime of “SinkDistComb” at network densities 0.3, 0.5, and 0.7, source node ratio 0.01 and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

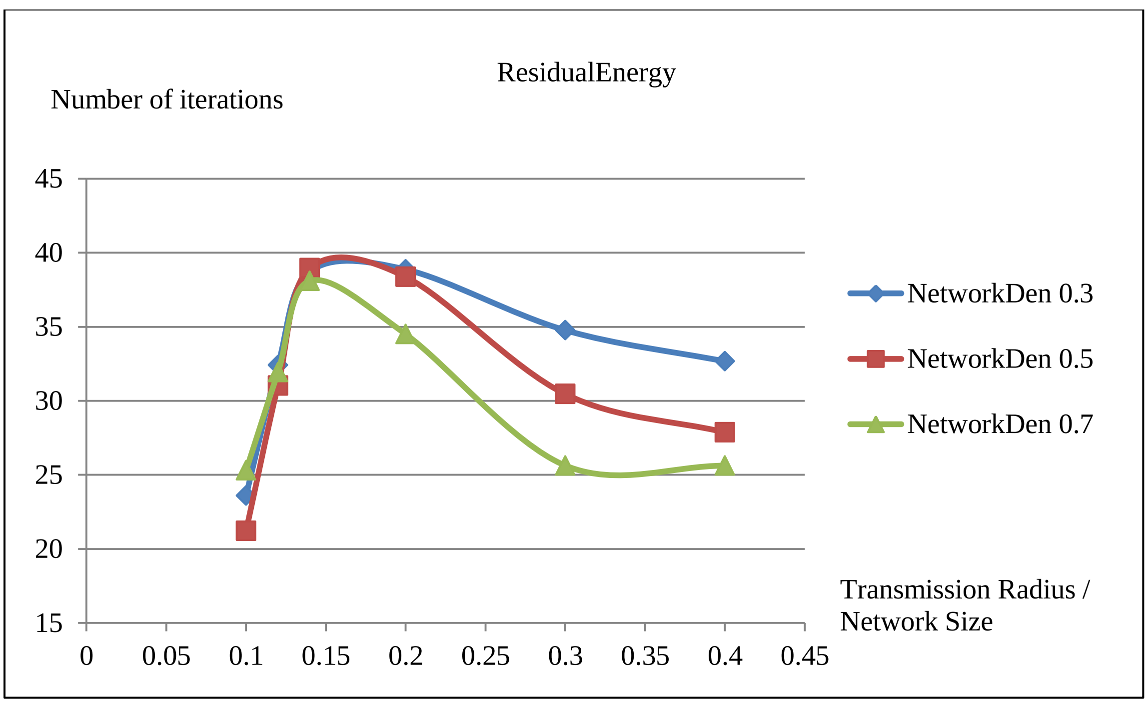

Figure 9. Network lifetime of “ResidualEnergy” at network densities 0.3, 0.5, and 0.7, source node ratio 0.01 and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

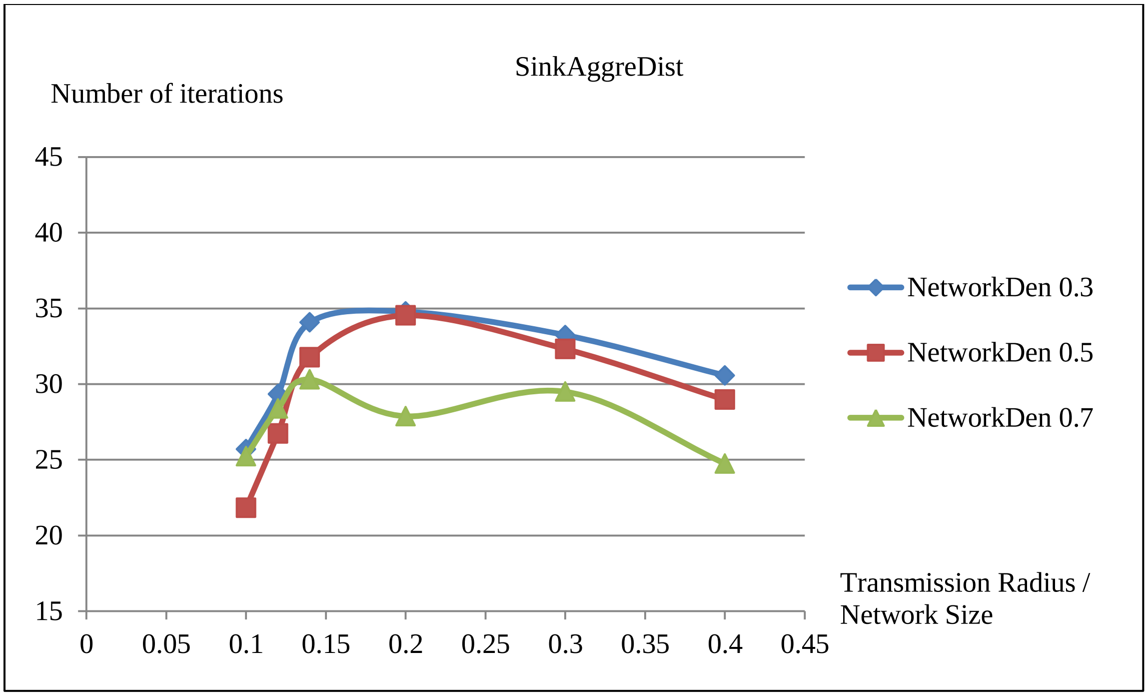

Figure 10. Network lifetime of “SinkAggreDist” at network densities 0.3, 0.5, and 0.7, source node ratio 0.01 and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

When source node ratio remains the same, increasing network density also increases the number of source nodes in the network. Figures 8 to 10 show that the network lifetimes of all three algorithms decline when the network density increases from 0.3 to 0.7. The algorithm “ResidualEnergy” maintains the same highest lifetime at all three densities while the other two algorithms’ decrease a little. The algorithm “SinkDistComb” has similar curve shapes at all three densities. The algorithm “SinkAggreDist” has the greatest lifetime decrease when the density grows.

Higher network density means more nodes in unit area. At the same transmission radius, a node has more neighborhood nodes. Nevertheless, since the number of source nodes increases, the network traffic is heavier. Figures 3 to 5 show the differences of the algorithms in handling greater network traffic and utilizing increased neighborhood nodes. Algorithm “ResidualEnergy” is able to choose the first node with most energy left in a larger candidate nodes pool. Therefore, it utilizes the increased nodes most efficiently. Algorithm “SinkDistComb” distributes energy consumption among nodes in multiple paths. “SinkAggreDist” may not be appropriate when the numbers of both sensor nodes and source nodes are large. This is because the frequent broadcasting need of aggregating nodes consumes more energy. Although “SinkAggreDist” encourages early aggregation to reduce traffic, the necessity to notify other nodes and synchronize the path with the aggregated nodes is costly.

It can be concluded that the all algorithms perform better at network density 0.3 than at two other higher densities. When the source node ratio is fixed and the network density grows, the increase of network traffic has a greater impact than the increase of neighborhood nodes. Therefore, the network lifetimes drop for all algorithms. Algorithm “ResidualEnergy” does the best job at maintaining stable network lifetime, followed by “SinkDistComb” and then “SinkAggreDist.

With same source node number. Figures 11, 12 and 13 illustrate the effect of network density with fixed source node number.

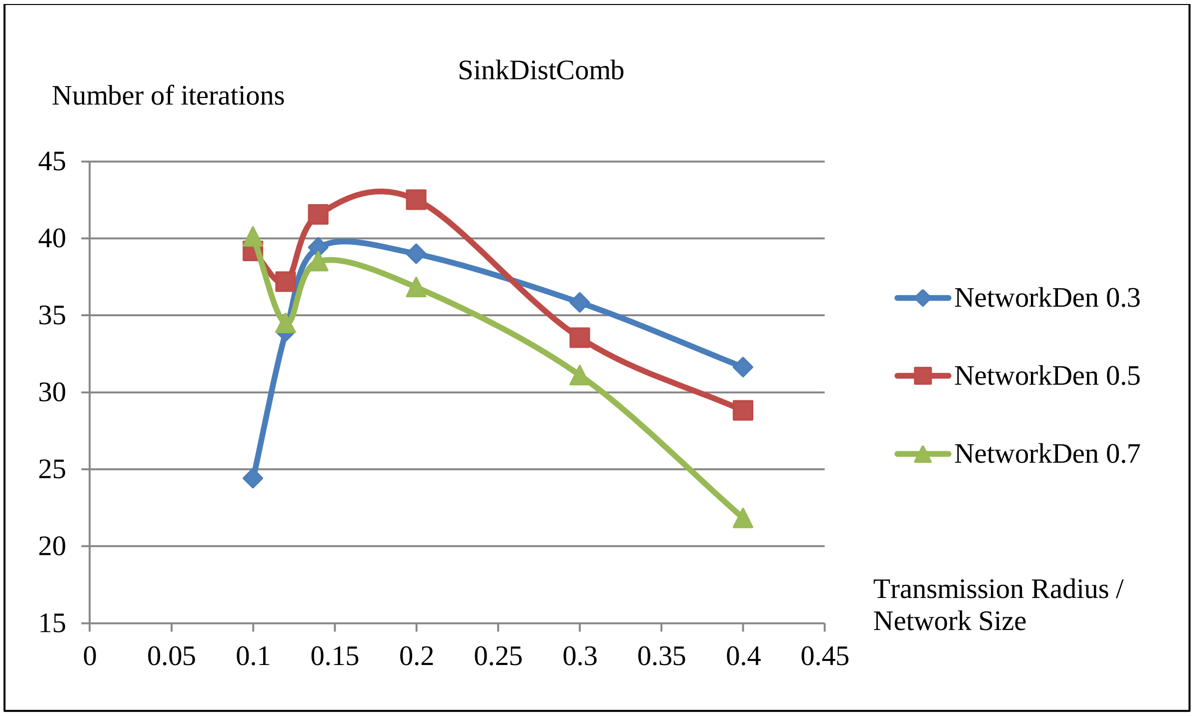

Figure 11. Network lifetime of "SinkDistComb" at network density 0.3, 0.5, and 0.7, source node number 8 and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

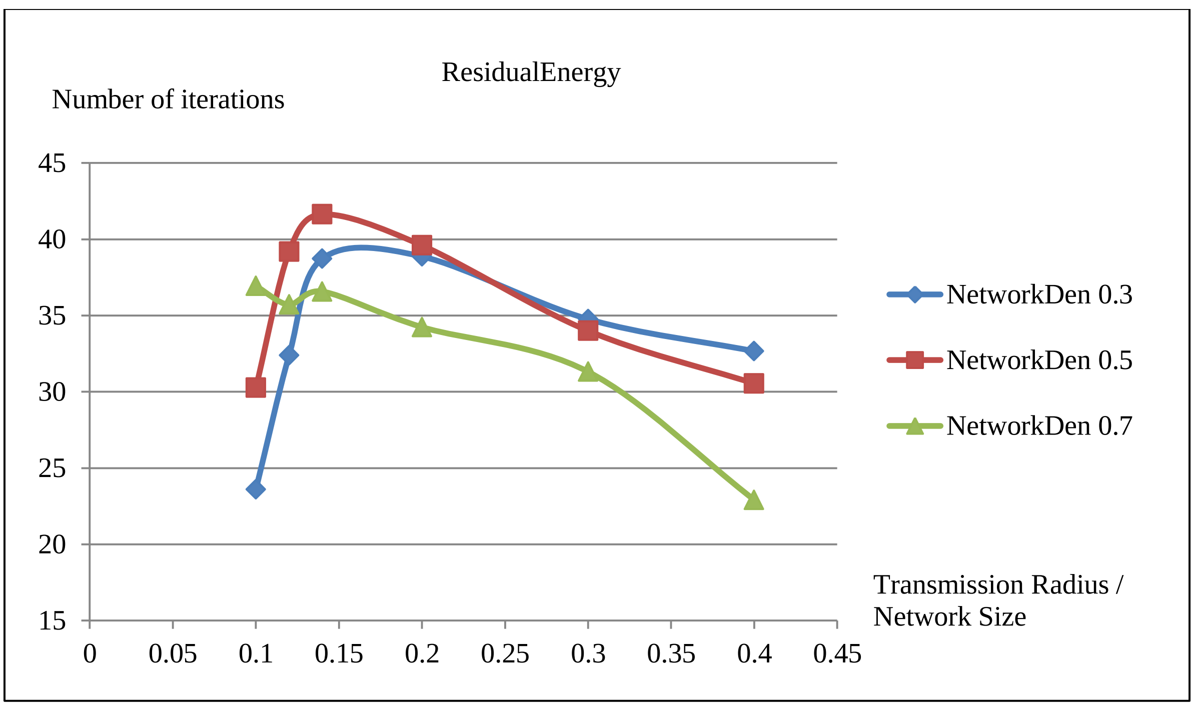

Figure 12. Network lifetime of "ResidualEnergy" at network density 0.3, 0.5, and 0.7, source node number 8 and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

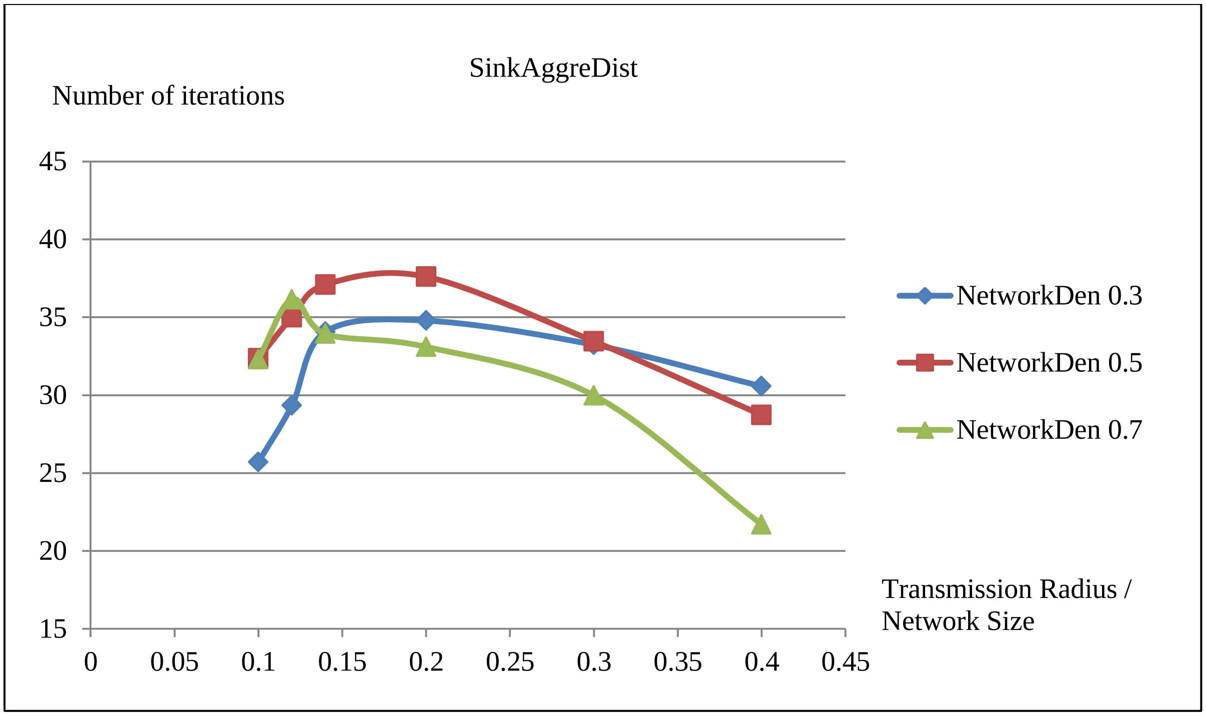

Figure 13. Network lifetime of "SinkAggreDist" at network density 0.3, 0.5, and 0.7, source node number 8 and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

Figures 11 to 13 show that when the network density increases and the source node number remains the same, the curves are left shifted. Between the transmission ratio of 0.1 and 0.4, the three algorithms have the best lifetime at network density 0.5 and the worse lifetime at density 0.7.

The curves are left shifted. Greater network density results in longer lifetime at shorter transmission range for density 0.3 and 0.5. In the figures, the curve shape of density 0.7 at transmission ratio below 0.1 is not shown. The longer lifetime can be explained by the higher availability of neighborhood nodes. It is needed to note that the reason is different from the one discussed in the section of Transmission Range, because the transmission radius is small and fixed. Higher network density enhances the flexibility of hop-by-hop routing assume total hop-count from the source node to the sink node is the same. The effect is illustrated by the left and middle drawings of Figure 14. When the transmission range becomes larger and the network size stays the same, the ratio of best candidate nodes over all neighborhood nodes becomes lower. These best candidate nodes are shared between different sending nodes (Figure 14 right). As a result, those few candidates are repeatedly used, which leads to shorter nodes' lifetime. When all the important nodes such as the ones within the transmission range of the sink node die, the network lifetime also ends. In another word, at larger transmission range, the flexibility of routing decreases and the advantages of the routing algorithms do not get applied.

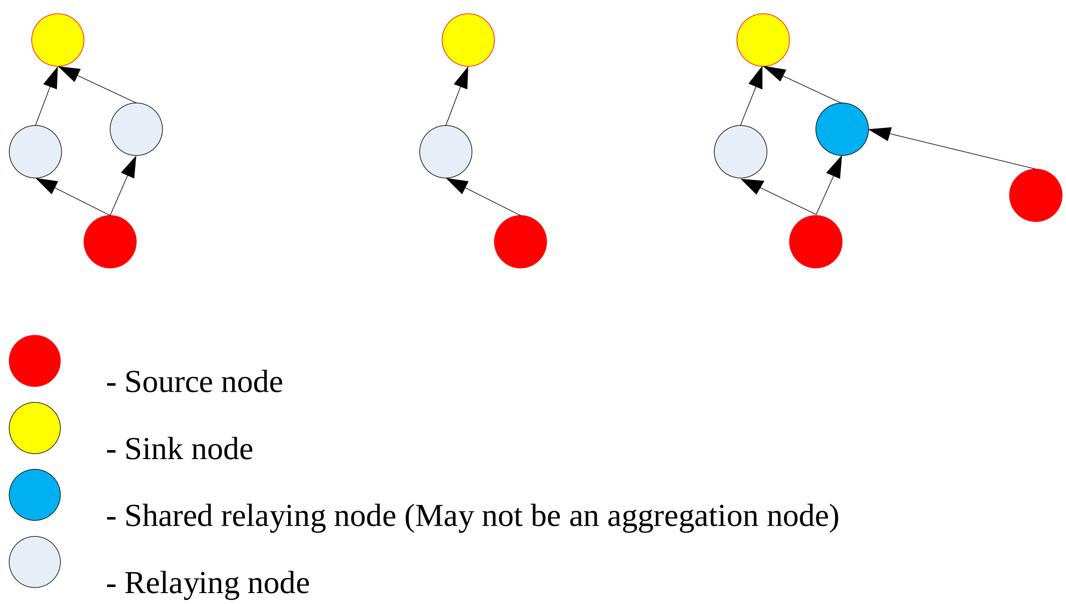

Figure 14. Effect of network density with the same source node number. (Left – higher network density at transmission radius 1; middle – lower network density at transmission radius 1; right - higher network density at transmission radius 1.5)

Source node ratio. The effect of increased source node ratio on network lifetime suggests the scalability of the ACO algorithms. Figure 15 shows the network lifetimes of the three algorithms at different source node ratios. Increased source node density results in decreased network lifetime. This is because the more source nodes, the more network traffic, and therefore the more energy consumption.

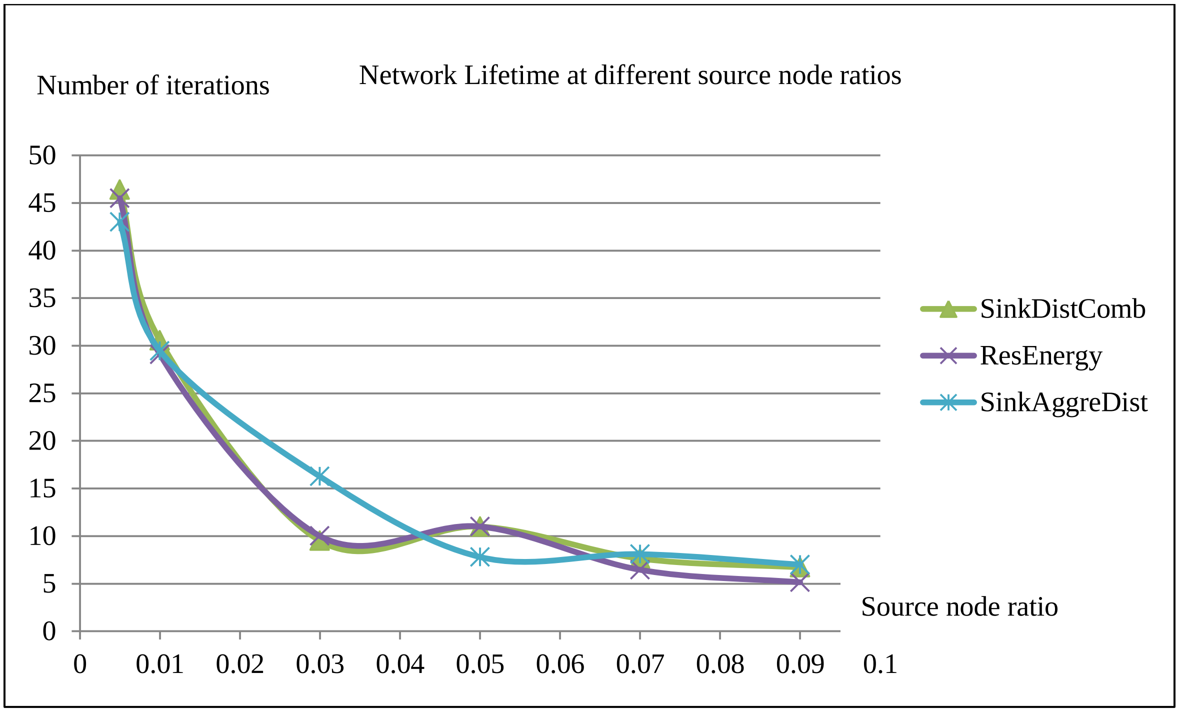

Figure 15. Network lifetime of the five algorithms with network density 0.3, transmission radius / network size 0.14, and source node ratio 0.005, 0.01, 0.03, 0.05, 0.07, and 0.09.

For algorithms “SinkDistComb” and “ResidualEnergy”, the network lifetime decreases rapidly when the source node ratio increases from 0.005 to 0.03. The lifetime becomes stable and decreases much slower from 0.03 above. For algorithm “SinkAggreDist”, the lifetime decrease is relatively mild. Before ratio 0.045, its lifetime is longer than the other two algorithms’. The shape of the curves can be explained by aggregation effect measured using the equation:

Aggregation times / Delivered package number

The effect of aggregation decreases quickly when the source node ratio increases from 0.01 to 0.03. The number of packets delivered to the sink node increases because there are more source nodes. Fewer aggregations are performed during the routing process. Therefore the aggregation effect declines. When ratio of source node is even higher, the number of packets delivered to the sink node becomes stable because in one aggregation more packets are aggregated into one packet. The aggregation algorithms are able to reduce the number of packets effectively during the routing process. As a consequence the lifetime curves decrease much slower.

Algorithm “SinkAggreDist” has better performance because it enbourages early aggregation. When the network load is heavier due to increased source nodes, “SinkAggreDist” reduces the traffic more effectively than the other two algorithms. As a result, it has better scalability than other ACO algorithms proposed.

Hop-count delay. The average number of hop-counts in solution paths reflects the general delay of the routing process and the quality of the paths. Figures 16, 17, and 18 compare the performance of the five algorithms and suggest the impact of transmission range and network density on the delay. Small delay is desired for timely delivery of packets.

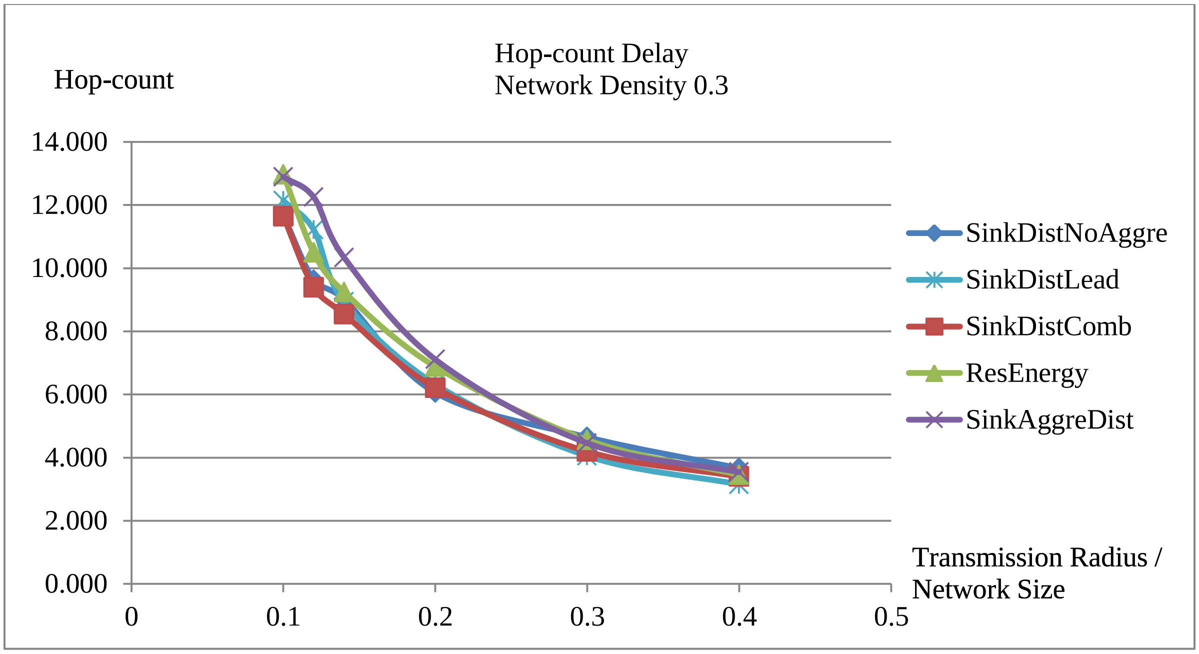

Figure 16. Hop-count delay of the five algorithms with network density 0.3, source node ratio 0.01, and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

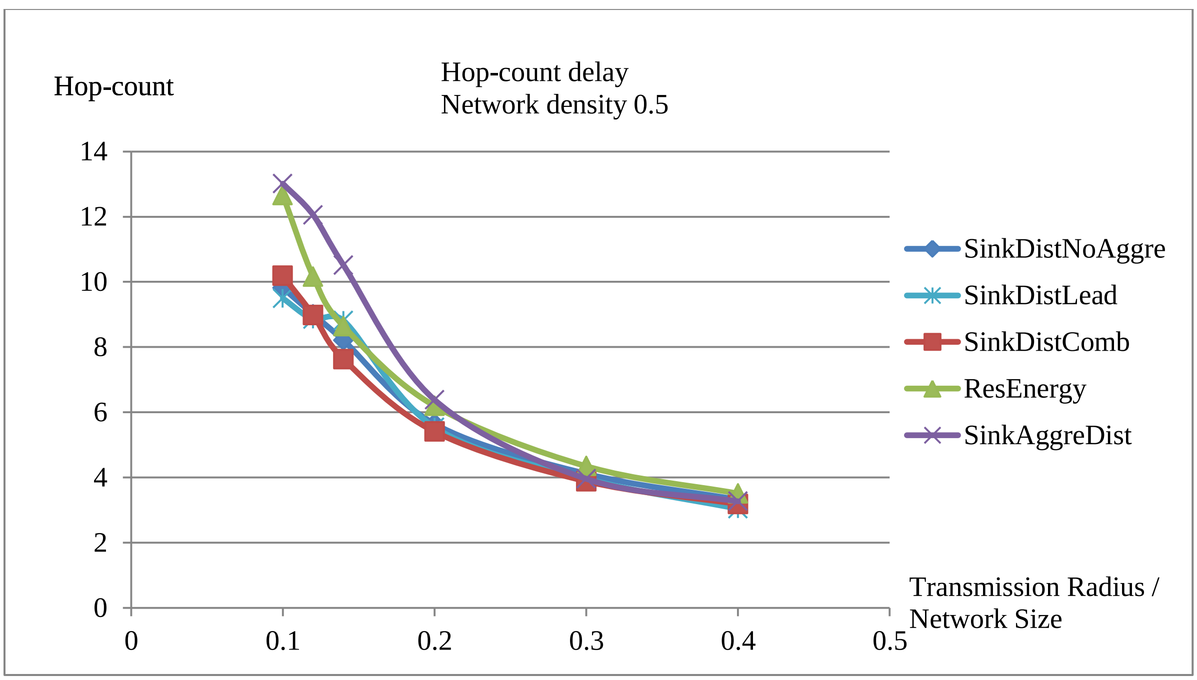

Figure 17. Hop-count delay of the five algorithms with network density 0.5, source node ratio 0.01, and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

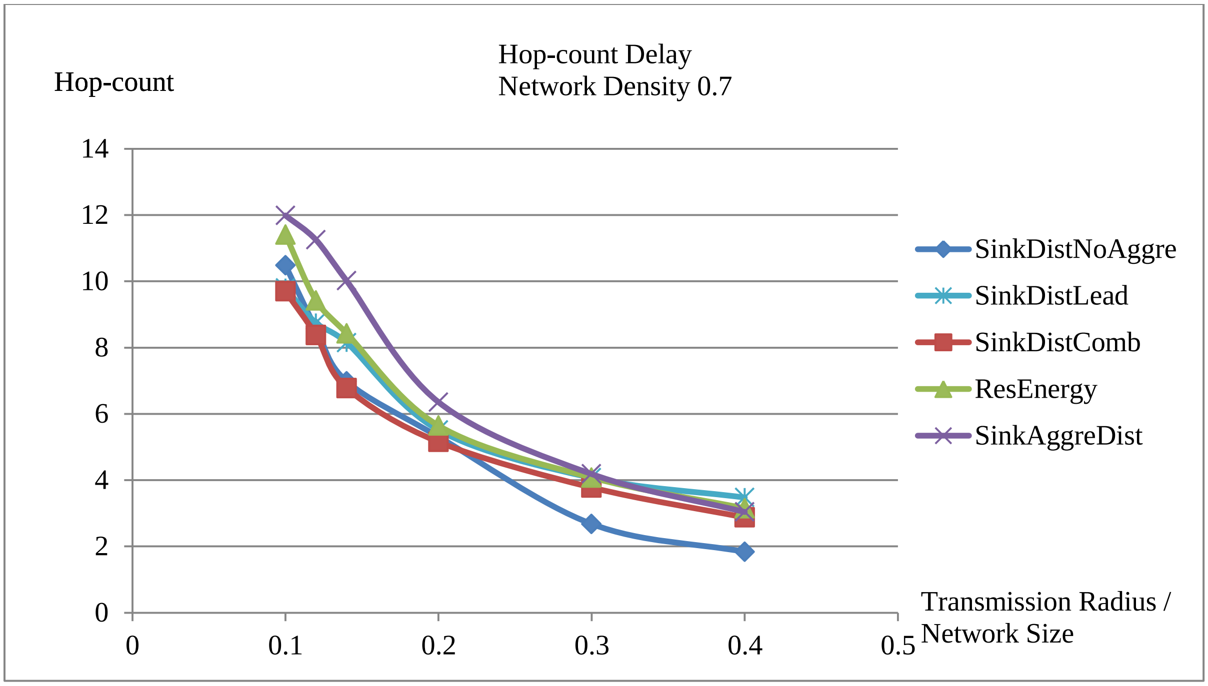

Figure 18. Hop-count delay of the five algorithms with network density 0.7, source node ratio 0.01, and transmission radius / network size 0.1, 0.12, 0.14, 0.2 and 0.4.

Figures 16 to 18 show that the five algorithms have small differences in hop-count delay, so the quality of their solution paths is similar. This is desirable because they prove that algorithms generating longer network lifetime also yield short delay. By comparing figures 3 to 5 with figures 16 to 18, it is found that the results of network lifetime and hop-count delay are not closely related. Although trade-off exists between the two performance metrics, efficient resources utilization can weaken the effect. The better performing algorithms use more sensor nodes to transmit data than traditional algorithms.

It is observed that “SinkDistComb”, “SinkDistNoAggre”, and “SinkDistLead” generate very close hop-counts. This result is expected because the three algorithms use the same heuristic. With in-network aggregation, “SinkDistComb” and “SinkDistLead” have much less energy consumption and network traffic than “SinkDistNoAggre”, therefore longer network lifetime. “SinkDistNoAggre” and “SinkDistLead” are greedy algorithms use the local optimal solution. By exploring multiple local optimal solutions, algorithm “SinkDistComb” has better chance to find the global optimal path solution. In general, “SinkDistComb” also has shorter delay than “ResidualEnergy” and “SinkAggreDist”. Although the latter two algorithms generate good lifetime result, they pay the price of slightly higher hop-count delay. “ResidualEnergy” does more routing in order to distribute energy consumption. “SinkAggreDist” does more routing to reduce packets in the network. However, “SinkAggreDist” consumes more energy to propagate the aggregating node information, therefore it has shorter network lifetime than the other two algorithms.

The effect of transmission range and network density on the hop-count delay for algorithms “SinkDistComb”, “ResidualEnergy”, and “SinkAggredist” is discussed in the following sections.

Transmission range. It is found from figures 16 to 18 that the delay decreases smoothly with the increase of transmission range. This is different from the result of network lifetime, which increases first and then decreases (figures 3 to 5). The high hop-count at short transmission range is because small relaying steps require more hop-by-hop routing to complete the whole transmission path. The network lifetime benefits from data aggregation and maximizes at certain transmission range but hop-count delay is not affected. When the range is relatively long, the source node can reach the sink node in a few steps no matter which routing path is selected. As a result, hop-count is small and stable at longer range.

Network density. The figures 16 to 18 suggest that increasing network density with fixed source node ratio at the same transmission radius results in decreased hop-count delay. This is caused by the higher density of surrounding nodes within the broadcaset area of a sensor node. The sending node has higher probability to transmit the packet to a further neighborhood node . Therefore, the transmission range is efficiently used. Instead, if the network density is low, the sending node has higher probability to send the packet to a node in the mid way and select the next relaying node from there. As a result, more hops are needed to deliver the packet to the destination node.

It is also found that as network density increases, the hop-count delay of "SinkDistComb" drops more than those of "ResidualEnergy" and "SinkAggreDist". This can be explained by the heuristic used. "SinkDistComb" uses single heuristic (distance to the sink node), thus it is affected more by network density than the algorithms using two combined heuristics.

Conclusion

This thesis discusses the effect of Ant-Colony Aggregation Algorithms with different heuristics and node selection rules on the network lifetime and hop-count delay of Wireless Sensor Network. It explains the importance of network lifetime as a major performance metric of WSN. Accordingly, it proposes three algorithms: “SinkDistComb”, “ResidualEnergy”, and “SinkAggreDist”, and compares the performance of the proposed algorithms and traditional algorithms. In addition, the effect of configuration parameters on the performance metrics is discussed.

Aggregation algorithms are proved to have much longer network lifetime than the non-aggregation algorithm. Multi-path algorithm “SinkDistComb” generated longer lifetime than the single-path algorithm “SinkDistLead”. The simulation results show that “ResidualEnergy” and “SinkDistComb” are more advantageous in extending network lifetime than “SinkAggreDist”, “SinkDistLead” and “SinkDistNoAggre”. “SinkAggreDist” provides better scalability for WSN. It is appropriate for a network with higher source node ratio. On the other hand, “SinkDistComb” generates the lowest hop-count delay, followed by “ResidualEnergy” and lastly “SinkDistNoAggre”.

In conclusion, “SinkDistComb” is the best proposed algorithm because it generates long network lifetime and low hop-count delay. Combined heuristic factors can be used for various routing purposes of WSN. The fact of nodes’ residual energy can be utilized in extending network lifetime. Repeatedly using the best candidate pool is also helpful for the same purpose with less computation overhead. Besides, it generates less delay than the residual energy. The extent of trade-off between lifetime and delay can be explored and utilized to achieve the application goal of the Wireless Sensor Network.

In the future, the heuristic of ACO algorithm can include spatial-correlation in different packets to enhance data integrity during aggregation process when multiple monitoring purposes are needed for the Wireless Sensor Network. The data quality of the packets received by sink node can be studied and different kinds of trade-off between performance metrics can be explored.

References

Abdelzaher, T., He, T., & Stankovic, J. (2004). Feedback control of data aggregation in sensor networks. *Proceedings of the 43rd IEEE Conference on Decision and Control, 2,* 1490 – 1495. doi: 10.1109/CDC.2004.1430254

Ahn, G., Hong, S. G., Miluzzo, E., Campbell, A. T., & Cuomo, F. (2006). Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks. *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems,* 293 – 306. doi: 10.1145/1182807.1182837

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks, 38(4),* 393 – 422.

Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications, 11(6),* 6 – 28. doi: 10.1109/MWC.2004.1368893

Al-Khdour, T., & Baroudi, U. (2007). A generalized energy-aware data centric routing for wireless sensor network. *Proceedings of the 2007 IEEE International Conference on Signal Processing and Communications*, 117 – 120. doi: 0.1109/ICSPC.2007.4728269

Al-khdour, T., & Baroudi, U. (2007). An entropy-based throughput metric for fairly evaluating WSN routing protocols. *Proceedings of 2007 IEEE International Conference on Network Protocols,* 342 – 343. doi: 10.1109/ICNP.2007.4375872

Ant colony optimization algorithms. (2010). Retrieved Nov 6th, 2011, from http://en.wikipedia.org/wiki/Ant_colony_optimization

Bai, F., & Jamalipour, A. (2008). Performance evaluation of optimal sized cluster based wireless sensor networks with correlated data aggregation consideration. *Proceedings of the 33rd IEEE Conference on Local Computer Networks,* 244 – 251. doi: 10.1109/LCN.2008.4664176

Benson, J. P., Roedig, U., Barroso, A., & Sreenan, C. J. (2007). On the effects of Aggregation on Reliability in Sensor Networks. *Proceedings of the 65th IEEE Vehicular Technology Conference*, 145 – 149. doi: 10.1109/VETECS.2007.42

Bezenchek, A., Rafanelli, M., & Tininini, L. (1996). A data structure for representing aggregate data. *Proceedings of the 8th International Conference on Scientific and Statistical Database Systems*, 22 – 31. doi: 10.1109/SSDM.1996.505912

Bisdikian, C. (2007). On sensor sampling and quality of information: a starting point. *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 279 – 284. doi: 10.1109/PERCOMW.2007.88

Bista, R., Jo, K. J., & Chang, J. W. (2009). A new approach to secure aggregation of private data in wireless sensor networks. *Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing,* 394 – 399. doi: 10.1109/DASC.2009.77

Blasch, E., & Plano, S. (2005). DFIG Level 5 (User Refinement) issues supporting Situational Assessment Reasoning. *Proceedings of the 8th International Conference on Information Fusion 1*, xxxv – xliii. doi: 10.1109/ICIF.2005.1591828

Blass, E. O., Wilke, J., & Zitterbart, M. (2006). A security-energy trade-off for authentic aggregation in sensor networks. *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks*, 135 – 137. doi: 10.1109/WIMESH.2006.288629

Boukerche, A., Cheng, X., & Linus, J. (2005). A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks 11(5)*, 619 – 635. doi: 10.1007/s11276-005-3517-6

Boulis, A., Ganeriwal, S., & Srivastava, M. B. (2003). Aggregation in sensor networks: an energy-accuracy trade-off. *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications,* 128 – 138. doi: 10.1109/SNPA.2003.1203363

Chang, J., & Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking, 12(4)*, 609-619. doi: 10.1109/TNET.2004.833122

Chen, G., Guo, T., Yang, W., & Zhao, T. (2006). An improved ant-based routing protocol in Wireless Sensor Networks. *Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 1 – 7.

Chen, K., Huang, J., & Hsiao, C. (2009). CHIRON: an energy-efficient chain-based hierarchical routing protocol in wireless sensor networks. *Proceedings of the 8th Wireless Telecommunications Symposium*, 1 – 5. doi: 10.1109/WTS.2009.5068960

Chen, S., & Zhang, Z. (2006). Localized algorithm for aggregate fairness in wireless sensor networks. *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, 274 – 285. doi:10.1145/1161089.1161121

Das, S., Singh, G., Gosavi, S., & Pujar, S. (2003). Ant colony algorithm for data-centric routing in sensor networks. *Proceedings of the Joint Conference on Information Sciences*, 1565 – 1568.

Deb, B., Bhatnagar, S., & Nath, B. (2003). ReInForM: reliable information forwarding using multiple Paths in sensor networks. *Proceedings of the 28th IEEE Int'l Conf. on Local Computer Networks*, 406-415. doi: 10.1109/LCN.2003.1243166

Deshpande, A. (2009). Data Compression in Sensor Networks. Retrieved June 7th, 2010, from http://www.cs.umd.edu/~amol/papers/compression.pdf

Di Bacco, G., Melodia, T., & Cuomo, F. (2004). A MAC protocol for delay-bounded applications in wireless sensor networks. *Proceedings of the 3rd Mediterranean Ad Hoc Networking.*

Ding, M., Cheng, X., & Xue, G. (2003). Aggregation tree construction in sensor networks. *Proceedings of the 58th IEEE Vehicular Technology Conference*, *4*, 2168 – 2172. doi: 10.1109/VETECF.2003.1285913

Dorigo, M. (1992). Optimization, learning and natural algorithms. *Doctoral dissertation*, Politecnico di Milano, Italie.

Dulman, S., Nieberg, T., Wu, J., & Havinga, P. (2003). Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. *Proceedings of the 2003 IEEE Wireless Communications and Networking, 3,* 1918 – 1922. doi: 10.1109/WCNC.2003.1200680

Erramilli, V., Malta, I., & Bestavros, A. (2004). On the interaction between data aggregation and topology control in wireless sensor networks. *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks,* 557 – 565. doi: 10.1109/SAHCN.2004.1381958

Estrin, D., Sayeed, A., & Srivastava, M. (2002). Mobicom 2002 Tutorial "Wireless Sensor Networks". Retrieved June 28th, 2010, from http://nesl.ee.ucla.edu/tutorials/mobicom02

Fasolo, E., Rossi, M., Widmer, J., & Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications, 14(2),* 70 – 87. doi: 10.1109/MWC.2007.358967

Galluccio, L., Campbell, A. T., & Palazzo, S. (2005). Concert: aggregation- based congestion control for sensor networks. *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems,* 274 – 275. doi:10.1145/1098918.1098951

Galluccio, L., & Palazzo, S. (2009). End-to-end delay and network lifetime analysis in a wireless sensor network performing data aggregation. *Proceedings of the IEEE Global Telecommunications Conference,* 1 – 6. doi: 10.1109/GLOCOM.2009.5425794

Galluccio, L., Palazzo, S., & Campbell, A. T. (2008). Efficient data aggregation in wireless sensor networks: An entropy-driven analysis. *Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,* 1 – 6. doi: 10.1109/PIMRC.2008.4699915

Galluccio, L., Palazzo, S., & Campbell, A. T. (2009). Modeling and designing efficient data aggregation in wireless sensor networks under entropy and energy bounds. *International Journal of Wireless Information Networks, 16,* 175 – 183. doi: 10.1007/s10776-009-0107-z

Guo, L., Ai, C., Wang, X., Cai, Z. & Li, Y. (2009). Real time clustering of sensory data in wireless sensor networks. *Proceedings of the 28th IEEE International Performance Computing and Communications Conference*, 33 – 40. doi: 10.1109/PCCC.2009.5403841

Haenggi, M. (2003). Energy-balancing strategies for wireless sensor networks. *Proceedings of the 2003 International Symposium on Circuits and Systems, 4,* 828 – 831. doi: 10.1109/ISCAS.2003.1206348

Heinzelman, W. R., Kulik, J., & Balakrishnan, H. (1999). Adaptive protocols for information dissemination in wireless sensor networks. *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 174 – 185.

Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences,* 10. doi: 10.1109/HICSS.2000.926982

Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *Proceedings of the IEEE Transactions on Wireless Communications, 1(4),* 660 – 670. doi: 10.1109/TWC.2002.804190

Huang, S., & Jan, R. (2004). Energy-aware, load balanced routing schemes for sensor networks. *Proceedings of 10th International Conference on Parallel and Distributed Systems*, 419 – 425. doi: 10.1109/ICPADS.2004.1316122

Intanagonwiwat, C., Estrin, D., Govindan, R., & Heidemann, J. (2002). Impact of network density on data aggregation in wireless sensor networks. *Proceedings of 22nd International Conference on Distributed Computing Systems,* 457 – 458. doi: 10.1109/ICDCS.2002.1022289

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidenmann, J., Siva, F. (2003). Directed diffusion for wireless sensor networking. *ACM/IEEE Transactions on Networking*, *11(1),* 2 – 16.

ISO/IEC Guide 99-12. (2007). *International Vocabulary of Metrology Basic and General Concepts and Associated Terms, ISO*.

Jagyasi, B. G., Dey, B. K., Merchant, S. N., & Desai, U. B. (2006). Weighted aggregation acheme with lifetime-accuracy tradeoff in wireless sensor network. *Proceedings of the 4th International Conference on Intelligent Sensing and Information*, 31 – 36. doi: 10.1109/ICISIP.2006.4286055

Joo, C., & Shroff, N. B. (2008). On the delay performance of in-network aggregation in lossy wireless sensor networks. *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing,* 483 – 490. doi: 10.1109/ALLERTON.2008.4797597

Jung, S. M., Han, Y. J., & Chung, T. M. (2007). The concentric clustering scheme for efficient energy consumption in the PEGASIS. *Proceedings of the 9th International Conference on Advanced Communication Technology, 1,* 260 – 265.

Kansal, A., & Zhao, F. (2009). Data fusion in sensor networks. Retrieved June 7th, 2010, from http://www.springerlink.com/content/n051593366l77776/fulltext.html

Khanna, G., Bagchi, S., & Wu, Y. (2004). Fault tolerant energy aware data dissemination protocol in sensor networks. *Proceedings of the International Conference on Dependable Systems and Networks,* 795 – 804. doi: 10.1109/DSN.2004.1311950

Kimura, N., & Latifi, S. (2005). A survey on data compression in wireless sensor networks. *Proceedings of the International Conference on Information Technology: Coding and Computing, 2,* 8 – 13. doi: 10.1109/ITCC.2005.43

Kollios, G., Byers, J., Considine, J., Hadjielefttheriou, M., & Li, F. (2005). Robust aggregation in sensor networks. *IEEE Data Engineering Bulletin, 28(1),* 26 – 32.

Krishnamachari, L., Estrin, D., & Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. *Proceedings of 22nd International Conference on Distributed Computing Systems Workshops,* 575 – 578. doi: 10.1109/ICDCSW.2002.1030829

Kulik, J., Heinzelman, W. R., & Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks, 8,* 169 –185.

Kumar, M., Verma, S., & Singh, P. P. (2008). Clustering approach to data aggregation in wireless sensor networks. *Proceedings of the 16th IEEE International Conference on,Networks,* 1 – 6. doi: 10.1109/ICON.2008.4772566

Lee, S., Bhattacharjee, B., & Banerjee, S. (2005). Efficient geographic routing in multihop wireless networks. *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 230 – 241.

Li, Q., Aslam, J., & Rus, D. (2001). Hierarchical Power-aware Routing in Sensor Networks. *Proceedings of the DIMACS Workshop on Pervasive Networking*.

Li, W., Bandai, M., & Watanabe, T. (2010). Tradeoffs among delay, energy and accuracy of partial data aggregation in wireless sensor networks. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications,* 917 – 924. doi: 10.1109/AINA.2010.137

Liang, X., Li, W., Gulliver, T. A. (2007). An energy-efficient MAC protocol exploiting the tree structure in wireless sensor networks. *Proceedings of the IEEE Military Communications Conference*, 1 – 7. doi: 10.1109/MILCOM.2007.4455331

Liao, W., Kao Y., & Fan, C. (2007). An ant colony algorithm for data aggregation in wireless sensor networks. *International Conference on Sensor Technologies and Applications,* 101 – 106.

Lin, F. Y., Yen, H., Lin, S., & Wen, Y. (2006). MAC aware energy-efficient data-centric routing in wireless sensor networks. *Proceedings of the IEEE International Conference on Communications, 8,* 3491 – 3496. doi: 10.1109/ICC.2006.255613

Lindsey, S., Raghavendra, C. S. (2002). PEGASIS: power-efficient gathering in sensor information systems. *Proceedings of the IEEE Aerospace Conference, 3*, 1125 – 1130. doi: 10.1109/AERO.2002.1035242

Lu, X., Spear, M., Levitt, K., & Wu, S. F. (2008). Non-uniform entropy compression for uniform energy distribution in wireless sensor networks. *Proceedings of the 2nd International Conference on Sensor Technologies and Applications,* 723 – 731. doi: 10.1109/SENSORCOMM.2008.98

Luo, H., Luo, J., Liu, Y., & Das, S. (2005). Energy efficient routing with adaptive data fusion in sensor Networks. P*roceedings of the 3rd ACM/SIGMOBILE Workshop on Foundations of Mobile Computing*. doi:10.1145/1080810.1080824

Madden, S., Franklin, M. J., Hellerstein, J. M., & Hong, W. (2002). TAG: a tiny aggregation service for ad-hoc sensor networks. *The 5th Symposium on Operating Systems Design and Implementation.*

Mahimkar, A., & Rappaport, T. S. (2004). SecureDAV: a secure data aggregation and verification protocol for sensor networks. *Proceedings of the IEEE Global Telecommunications Conference*, *4*, 2175 – 2179. doi: 10.1109/GLOCOM.2004.1378395

Manjeshwar A., & Agarwal, D. P. (2001). TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, 2009 – 2015. doi: 10.1109/IPDPS.2001.925197

Manjeshwar A., & Agarwal, D. P. (2002). APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *Parallel and Distributed Processing Symposium. Proceedings International*, 195 – 202. doi: 10.1109/IPDPS.2002.1016600

Manjhi, A., Nath, S., & Gibbons, P. B. (2005). Tributaries and deltas: efficient and robust aggregation in sensor network streams. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 287 – 298. doi: 10.1145/1066157.1066191

Misra, R., & Mandal, C. (2006). Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks. *Proceedings of IFIP International Conference Wireless and Optical Communications Networks,* 5. doi: 10.1109/WOCN.2006.1666600

Motegi, S., Yoshihara, K., & Horiuchi, H. (2006). DAG based in-network aggregation for sensor network monitoring. *International Symposium on Applications and the Internet*, 292 – 299. doi: 10.1109/SAINT.2006.20

Nath, S., Gibbons, P. B., Seshan, S., & Anderson, Z. R. (2004). Synopsis diffusion for robust aggregation in sensor networks. *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. doi: 10.1145/1031495.1031525

Padmanabh, K., & Vuppala, S. K. (2009). An adaptive data aggregation algorithm in wireless sensor network with bursty source. *Wireless Sensor Network, 3*, 222 – 232. doi: 10.4236/wsn.2009.13029

Pattem, S., Krishnamachari, B., & Govindan, R. (2004). The impact of spatial correlation on routing with compression in wireless sensor networks. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 28 – 35. doi: 10.1145/984622.984627

Pham, T., Kim, E. J., & Moh, M. (2004). On data aggregation quality and energy efficiency of wireless sensor network protocols - extended summary. *Proceedings of the 1st International Conference on Broadband Networks*, 730 – 732. doi: 10.1109/BROADNETS.2004.51

Pottie, G. J., & Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM, 43(5),* 51–58.

Shah, R. C., & Rabaey, J. M. (2002). Energy aware routing for low energy ad hoc sensor networks. *Proceedings of the 2002 IEEE Wireless Communications and Networking Conference, 1,* 350 – 355. doi: 10.1109/WCNC.2002.993520

Solis, I., & Obraczka, K. (2005). Isolines: energy-efficient mapping in sensor networks. *Proceedings of the 10th IEEE Symposium on Computers and Communications,* 379 – 385. doi: 10.1109/ISCC.2005.88

Solis, I., & Obraczka, K. (2006). In-network aggregation trade-offs for data collection in wireless sensor networks. *International Journal of Sensor Networks, 1,* 200 – 212. doi: 10.1504/IJSNET.2006.012035

Tabassum, N., Mamun, Q. E. K. M., & Urano, Y. (2006). COSEN: a chain oriented sensor network for efficient data collection. *Proceedings of the 3rd International Conference on Information Technology: New Generations,* 262 – 267. doi: 10.1109/ITNG.2006.44

Tabibzadeh, M., Sarram, M., & Adibnia, F. (2009). Hybrid routing protocol for prolonged network lifetime in large scale wireless sensor network. *Proceedings of the 2009 International Conference on Information and Multimedia Technology,* 179 – 183. doi: 10.1109/ICIMT.2009.64

Tilak, S., Abu-Ghazaleh, N., & Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review, 6(2),* 28 – 36.

Wang, H., & Luo, N. (2010). An improved ant-based algorithm for data aggregation in wireless sensor networks. *Proceedings of the 2010 International Conference on Communications and Mobile Computing, 3,* 239 – 243. doi: 10.1109/CMC.2010.183

Wang, H., Pottie, G., Yao K., & Estrin, D. (2004). Entropy-based sensor selection heuristic for target localization. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks,* 36 – 45. doi: 10.1109/IPSN.2004.1307321

Wang, J., He, H., Chen, B., Chen Y., & Guan, T. (2009). Data aggregation and routing in wireless sensor networks using improved ant colony algorithm. *Proceedings of the 2009 International Forum on Computer Science-Technology and Applications, 3,* 215 – 218. doi: 10.1109/IFCSTA.2009.292

Wireless sensor network. (2010). Retrieved June 4th, 2010, from http://en.wikipedia.org/wiki/Wireless_sensor_network

Xia, S., & Wu, S. (2009). Ant colony-based energy-aware multipath routing algorithm for wireless sensor networks. *Proceedings of the 2nd International Symposium on Knowledge Acquisition and Modeling,* 198 – 201.

Xia, S., Wu, S., & Ni, J. (2009). A new energy-efficient routing algorithm based on ant colony system for wireless sensor networks. *Proceedings of the 4th International Conference on Internet Computing for Science and Engineering,* 176 – 180.

Xiong, Z., Liveris, A. D., & Cheng, S. (2004). Distributed source coding for sensor networks. *IEEE Signal Processing Magazine, 21(5),* 80 – 94. doi: 10.1109/MSP.2004.1328091

Xu, Y., Heidemann, J., & Estrin, D. (2001). Geography-informed energy conservation for ad hoc routing. *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking,* 70 – 84.

Yang, J., Lin, Y., Xiong W., & Xu, B. (2009). Ant colony-based multi-path routing algorithm for wireless sensor networks. *Proceedings of the International Workshop on Intelligent Systems and Applications,* 1 – 4.

Yao, Y., & Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record, 31(3),* 9 – 18.

Yao, Y., & Gehrke, J. (2003). Query processing for sensor networks. *Proceedings of the 1st Biennial Conference Innovative Data Systems Research*.

Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Proceedings of the IEEE Transactions on Mobile Computing, 3(4),* 366 – 379. doi: 10.1109/TMC.2004.41

Yu, X., Mehrotra, S., & Venkatasubramanian, N. (2007). Sensor scheduling for aggregate monitoring in wireless sensor networks. *Proceedings of the 19th International Conference on Scientific and Statistical Database Management,* 24. doi: 10.1109/SSDBM.2007.42

Yu, Y., Krishnamachari, B., & Prasanna, V. K. (2004). Energy-latency tradeoffs for data gathering in wireless sensor networks. *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, 1,* 255. doi: 10.1109/INFCOM.2004.1354498

Zhou, B., Ngoh, L. H., Lee, B. S., & Fu, C. P. (2004). A hierarchical scheme for data aggregation in sensor network. *Proceedings of 12th IEEE International Conference on Networks, 2,* 525 – 529. doi: 10.1109/ICON.2004.1409223

Zhu, Y., Sundaresan, K., & Sivakumar, R. (2005). Practical limits on achievable energy improvements and useable delay tolerance in correlation aware data gathering in wireless sensor networks. *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks,* 328 – 339.

VITA

Meng Xie is a Master candidate in Computer Science at Texas State University-San Marcos, Texas. She is working under the supervision of Dr. Hongchi Shi. Her research area is wireless sensor networks. She entered the Graduate College of Texas State in June, 2009. Before that, she received her Master's degree of Food Science and Nutrition from University of Missouri-Columbia in 2005 and her Bachelor's degree of Food Science from Zhejiang University, China in 2003.

Permanent Address: 16057 NE 8th Sreet

Apartment #101

Bellevue, WA, 98008

Note: Permanent email address may be used in lieu of mailing address

This thesis was typed by Meng Xie.