HIERARCHICAL ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORKS

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Andrew LeBeau

September 2012
COMMITTEE MEMBERSHIP

TITLE: Hierarchical Routing Protocols in Wireless Sensor Networks

AUTHOR: Andrew LeBeau

DATE SUBMITTED: September 2012

COMMITTEE CHAIR: John Seng, Ph.D.

COMMITTEE MEMBER: Phillip Nico, Ph.D.

COMMITTEE MEMBER: Franz Kurfess, Ph.D.
Abstract

Hierarchical Routing Protocols in Wireless Sensor Networks

Andrew LeBeau

In recent wireless sensor network research, using a mobile data center (sink) to collect data has shown the ability to decrease the overall energy expenditure of the sensor field. Before the introduction of mobile sinks, many different routing protocols were developed under the assumption of a stationary sink. This work compares three such routing protocols, one hierarchical and two non-hierarchical, under the assumption of a mobile sink to determine which is best. The three protocols are tested against varying sink speed, node communication radius, and sensor field node populations. Different sink movement strategies and modifications to the routing protocols are also evaluated. This work shows that a modified Directed Diffusion, a non-hierarchical routing protocol, performs slightly better on average than a modified TEEN, a hierarchical routing protocol.
Contents

List of Tables vii
List of Figures viii

1 Introduction 1
  1.1 Applications of Wireless Sensor Networks .......................... 2
    1.1.1 VigilNet .............................................. 3
    1.1.2 AlarmNet .............................................. 3
    1.1.3 ZebraNet .............................................. 4
  1.2 Energy Efficiency and Routing Protocols .......................... 5
  1.3 Moving Sinks .............................................. 7
  1.4 Contribution .............................................. 9

2 Previous Work 10

3 Data-Centric Routing Protocols 12
  3.1 Flooding ................................................. 13
  3.2 Directed Diffusion ........................................ 13
    3.2.1 Directed Diffusion Modifications ...................... 15
  3.3 TEEN ....................................................... 16
    3.3.1 TEEN Modifications ................................... 17

4 Sink Movement Strategies 19
  4.1 Straight ................................................. 19
  4.2 Random ................................................... 20
  4.3 Biased Random .......................................... 22
  4.4 Predetermined .......................................... 23
5 Sensor Field Simulator

5.1 Simulator Assumptions ........................................ 26
5.2 Simulator Iterations ........................................... 27
5.3 Routing Protocols ............................................. 28
5.4 Movement Strategies .......................................... 28

6 Experiment Methodology ........................................... 29

6.1 Simulation Variables ........................................... 29
6.2 Evaluation Criteria ............................................ 30

7 Results .............................................................. 32

7.1 Testing Movement Strategy ................................... 32
  7.1.1 Directed Diffusion and Movement ...................... 33
  7.1.2 Flooding and Movement ................................. 35
  7.1.3 TEEN and Movement ................................... 38
  7.1.4 Movement Conclusion ................................... 42

7.2 Testing Modified Protocols ................................. 43
  7.2.1 Directed Diffusion ......................................... 43
  7.2.2 TEEN ........................................................ 51

7.3 Testing Hierarchical versus Non-Hierarchical ........... 63
  7.3.1 Hierarchical vs. Non-Hierarchical and Sink Speed ... 63
  7.3.2 Hierarchical vs. Non-Hierarchical and Node Communication Radius ..................................... 64
  7.3.3 Hierarchical vs. Non-Hierarchical and Node Count .... 67
  7.3.4 Hierarchical vs. Non-Hierarchical Conclusion ....... 69

8 Future Work ........................................................ 72

9 Conclusion ......................................................... 74

Bibliography ........................................................ 77
List of Tables

7.1 Table of Directed Diffusion Modification Test Results . . . . . . . . . . . 51
7.2 Table of TEEN Modification Test Results . . . . . . . . . . . . . . . . . . 62
7.3 Table of Hierarchical vs. Non-Hierarchical Test Results . . . . . . . 71
List of Figures

1.1 Basic Sensor Field ........................................... 2
1.2 Sensor Field With Moving Sink ............................. 7

3.1 Flooding ....................................................... 14
3.2 Directed Diffusion ........................................... 15
3.3 Directed Diffusion Modified ............................... 16
3.4 TEEN ......................................................... 18

4.1 Sink Movement Straight Path ............................. 20
4.2 Sink Movement Random Path .............................. 22
4.3 Sink Movement Predetermined Path ..................... 24

7.1 Graph of Directed Diffusion Movement Strategy Speed Test .... 33
7.2 Graph of Directed Diffusion Movement Strategy Radius Test .... 34
7.3 Graph of Directed Diffusion Movement Strategy Node Test .... 35
7.4 Graph of Flooding Movement Strategy Speed Test ........... 36
7.5 Graph of Flooding Movement Strategy Radius Test ........... 37
7.6 Graph of Flooding Movement Strategy Node Test ........... 38
7.7 Graph of TEEN Movement Strategy Speed Test .......... 39
7.8 Graph of TEEN Movement Strategy Radius Test .......... 40
7.9 Graph of TEEN Movement Strategy Node Test .......... 41
7.10 Graph of Directed Diffusion Modification Speed Test - Data Collection .................. 44
7.11 Graph of Directed Diffusion Modification Speed Test - Message Transmission ........................................ 45
7.12 Graph of Directed Diffusion Modification Speed Test - Product .................................................. 46
7.13 Graph of Directed Diffusion Modification Radius Test - Data Collection ........................................... 46
7.14 Graph of Directed Diffusion Modification Radius Test - Message Transmission .................................. 47
7.15 Graph of Directed Diffusion Modification Radius Test - Product .................................................. 48
7.16 Graph of Directed Diffusion Modification Node Test - Data Collection ........................................... 49
7.17 Graph of Directed Diffusion Modification Node Test - Message Transmission .................................. 49
7.18 Graph of Directed Diffusion Modification Node Test - Product .................................................. 50
7.19 Graph of TEEN Modification Speed Test - Data Collection ....................................................... 52
7.20 Graph of TEEN Modification Speed Test - Message Transmission .................................................. 53
7.21 Graph of TEEN Modification Speed Test - Product ................................................................. 54
7.22 Graph of TEEN Modification Radius Test - Data Collection ....................................................... 55
7.23 TEEN Modification Radius Test - Message Transmission ........................................................... 56
7.24 Graph of TEEN Modification Radius Test - Product ................................................................. 57
7.25 Graph of TEEN Modification Node Test - Data Collection ........................................................... 58
7.26 Graph of TEEN Modification Node Test - Message Transmission .................................................. 59
7.27 Graph of TEEN Modification Node Test - Product ................................................................. 60
7.28 Graph of TEEN Node Count Investigation .................................................................................. 62
7.29 Graph of Hierarchical vs. Non-Hierarchical Speed Test - Data Collection ...................................... 64
7.30 Graph of Hierarchical vs. Non-Hierarchical Speed Test - Message Transmission .................................. 65
7.31 Graph of Hierarchical vs. Non-Hierarchical Speed Test - Product .................................................. 66
7.32 Graph of Hierarchical vs. Non-Hierarchical Radius Test - Data Collection ...................................... 66
7.33 Graph of Hierarchical vs. Non-Hierarchical Radius Test - Message Transmission .............................. 67
7.34 Graph of Hierarchical vs. Non-Hierarchical Radius Test - Product .................................................. 68
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.35</td>
<td>Graph of Hierarchical vs. Non-Hierarchical Node Test - Data Collection</td>
<td>68</td>
</tr>
<tr>
<td>7.36</td>
<td>Graph of Hierarchical vs. Non-Hierarchical Node Test - Message Transmission</td>
<td>69</td>
</tr>
<tr>
<td>7.37</td>
<td>Graph of Hierarchical vs. Non-Hierarchical Node Test - Product</td>
<td>70</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

A wireless sensor network is a distribution of independent sensor nodes over an area. Each node is small, battery powered, senses and records information about its local environment, and can transmit its data via an RF transceiver. The number of nodes deployed in the sensor field can range from tens to thousands or more. Since the individual nodes have low processing power and limited energy resources, nodes transmit their data to a more powerful central base station (known as a sink) via radio transmission for long term storage and data analysis.

The sink is not likely within range of all the nodes in the sensor field, so to reach the sink, nodes transmit their data to other nodes which are closer to the sink. This pattern continues until the data reaches the sink and is a process known as multi-hop routing. Figure 1.1 shows an example of a sensor field with nodes further away from the sink that hop their messages over multiple messages to reach the sink. In this figure, the data from node 6 first transmits to node 4, then node 3 and node 1 and then on to the sink.

Nodes are usually deployed in an ad-hoc manner, meaning they have no prior
knowledge of their environment or position relative to other nodes, and are expected to self-organize themselves to transmit their data to the sink. This self-organization process and the algorithm used to get data messages to the sink is known as a routing protocol.

1.1 Applications of Wireless Sensor Networks

There are many applications for wireless sensor networks. This section illustrates three: VigilNet, a military surveillance system; AlarmNet, an assisted living facility emergency detection system; and ZebraNet, a wildlife habitat monitoring system.
1.1.1 VigilNet

VigilNet [6][13] is a military surveillance system with nodes designed to be deployed into a hostile region for a lifetime of 3 to 6 months. These nodes are designed to detect events of interest, including the presence of people, weapons, and vehicles. In VigilNet, the nodes are responsible for detecting, tracking, and classifying events of interest, mapping them to a geographical location, and routing the information back to the sink within an acceptable latency. Acceptable latency depends on the severity of the event detected.

To conserve energy, VigilNet uses many nodes that use little energy and act like tripwires to detect the occurrence of an event. When these nodes detect an event, they notify other higher-powered systems in the area, like infrared cameras, that record information about the event. This information is then passed to other classification nodes that classify the risk of the threat and manage transmission priority of that data back to the sink.

1.1.2 AlarmNet

AlarmNet [14][15] is a wireless sensor network designed to monitor patients living at large-scale assisted living facilities. In this system, a separate group of nodes is allocated to monitor each patient. Some nodes are mobile and connected directly to the patient and monitor vital statistics like heart rate and body temperature. Other nodes are placed statically within the living area of the patient to monitor for environmental conditions like motion, smoke, or gas. This data helps inform the healthcare provider if there is an emergency situation as well as track health and activity information about the patient over time.
Since AlarmNet is localized within a residential area, it can employ multiple stationary sinks, like a PC, that is connected to a central database and close to each of the group of nodes. Multiple sinks helps reduce energy expenditure because messages don’t have to travel as far to reach a sink. Additionally, some of the static sensor nodes can even be hardwired so that they do not have energy limitation and are directly connected to a LAN.

1.1.3 ZebraNet

Finally, ZebraNet [8] is an application designed to monitor the migratory patterns of zebras in central Kenya. The ZebraNet project attached collars to a sample size of wild zebras that roamed thousands of miles of Kenyan grasslands. These collars were designed to take regular GPS readings to track position and movement of the zebras. The collars were also fitted with radio transmitters to multi-hop data back to the sink. To replace the batteries in a collar, the zebra would need to be tranquilized. To minimize tranquilizations, collars needed to conserve energy so they could last at least one year.

Since the zebras could move over such a large area, the sink could not possibly remain stationary the whole time and remain in range of small, low-powered nodes. To solve this problem, the sink was attached to a research vehicle and driven through the Kenyan grasslands to periodically collect data. The research vehicles would not always be within range of any nodes, so the sensor field had to be outfitted with a routing protocol that would only route data at key times to conserve energy.
1.2 Energy Efficiency and Routing Protocols

Since the nodes in a sensor field are battery powered, minimizing the energy expenditure of the sensor field is key to ensuring that the sensor field can operate for the longest period of time possible. Much research has been done to develop different types of routing protocols and examine their effect on energy efficiency.

One such example of a routing protocol is Directed Diffusion [7]. Directed Diffusion uses a query-based system where the sink sends messages through the sensor field to request specific data from specific nodes. As this query message is propagated through the field, paths between nodes and the sink are established. If the node has the requested data in memory, it can transmit the data along the path that requires the fewest hops to get to the sink. Data is also sent along the non-optimal paths in case there are node failures along the optimal path, but at a lower transmission rates.

Directed Diffusion also introduced the concept of data aggregation. Data aggregation occurs when a node aggregates sensor data from multiple different nodes to include in a single message packet. Data aggregation also eliminates duplicate messages that come from the same node, which reduces the total number of messages in the system. Both query-based messaging and data aggregation aim to reduce the total number of messages sent through the system. Fewer messages spent means less energy expended and the lifetime of the sensor field is extended.

While Directed Diffusion aims to minimize the total number of messages sent, this approach can cause other problems in a wireless sensor network. If all nodes use the shortest path to transmit their data to the sink, several nodes in the field, particularly those in high density areas or in close proximity to the sink, will have
to transmit many more messages than those in the outskirts. Those high-traffic nodes will use significantly more power and run out of energy sooner than other nodes. The amount of time a sensor field can operate before a single node dies off is known as total network lifetime.

A modification to Directed Diffusion to address this problem was made in the Energy Aware Routing protocol [12]. Energy Aware Routing takes what Directed Diffusion started with, but instead of sending through multiple paths and biasing for the shortest path to the sink, Energy Aware Routing only transmits along one path, but picks that path randomly. Only picking one path reduces the energy expenditure, but adds risk of lost packets delaying transmission. Choosing different paths over time also distributes the energy cost over more nodes. Energy Aware Routing found a 21.5 percent total energy savings and a 44 percent increase in total network lifetime.

Another branch of routing protocol research has studied the effectiveness of grouping nodes into hierarchical clusters. Each cluster contains multiple nodes and in many ways works like its own independent sensor field. One of the first hierarchical routing protocols was Low-Energy Adaptive Clustering Hierarchy (LEACH) [2].

LEACH is a single-hop routing protocol that divides the sensor field into clusters and assigns a leader to each cluster. The sensor nodes within a cluster regulate their energy used to transmit data and only use as much energy as is required to reach the leader of the cluster. The leader then aggregates the data and expends a larger amount of energy to transmit the data directly to the sink. To increase total network lifetime, LEACH periodically rotates the leader responsibility to other nodes. LEACH researchers found that the optimal number of nodes to make cluster heads is 5 percent of the total number of nodes in the
Figure 1.2: This is an example sensor field with a moving sink. The paths the messages take to reach the sink must change over time.

sensor field.

Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [10] is another hierarchical routing protocol. Unlike LEACH, TEEN supports multi-hop routing. Individual nodes within clusters can multi-hop to the leader node, and the leader node aggregates data and can pass that data on to other clusters. Only the leader node can communicate to other clusters. Researchers found that TEEN uses fewer total messages than LEACH, and also increases the total network lifetime.

1.3 Moving Sinks

More recently, researches have experimented with the idea of moving the sink through the sensor field as part of the data collection process. Unlike the
individual nodes, the sink does not have energy constraints. Moving the sink through the field can help reduce the total number of messages sent through the system by getting closer to nodes on the outskirts of the system. As shown in Figure 1.2, the sink first starts in a position where only node 1 can communicate directly with the sink, but later it moves to a position where nodes 2, 4, and 5 are all within range of the sink.

Changing the position of the sink will also change the shortest path to the sink over time. This can help distribute the energy expenditure across nodes and increase total network lifetime. However, changing the shortest path to the sink over time can also produce problems for routing protocols like Directed Diffusion that try to optimize for the shortest path to the sink. Directed Diffusion takes time to discover the shortest path back to the sink and the sink may no longer be within range of that path by the time the messages propagate.

When looking at the effectiveness of moving sinks, a key area of study is how the sink should move to best optimize energy efficiency of the sensor field. Younis, Bangad and Akkaya [16] looked at a moving sink approach where the sink moves through the sensor field and stops in high-traffic areas. This approach did not compare this movement strategy against other movement strategies, but did find an increase in savings over a non-moving sink approach.

Other research has compared different movement strategies between each other. Luo and Hubaux [9] tested three different movement approaches. The first strategy tries to move the sink so that it visits every node with equal frequency. The second strategy moves the sink in a circle halfway through the radius of the sensor field. The third strategy moves the sink in a circle around the circumference of the field. That research found that moving around at the circumference yielded the best results.
Another work looked at how random sink movement through the field performs. Chatzigiannakis, Kinalis, and Kinalis [3] tried two random sink movement approaches, true random and biased random, and also looked at a predetermined movement strategy that moves around the periphery of the field, like Luc and Hubaux did. They found that using a random or biased random movement strategy with no multi-hop routing used the least energy, but required more time to collect all of the data in the field.

1.4 Contribution

Over the years, many different routing protocols have been developed. Many of these protocols assumed a stationary sink. As the concept of a moving sink has been introduced, the effectiveness of previous routing protocols under this new scenario is uncertain. This work looks at two different categories of previously developed routing protocols, hierarchical and non-hierarchical, and evaluates how well they perform against each other, and explores what modifications can be made to optimize these routing protocols in a sensor field with a mobile sink.
Chapter 2

Previous Work

While no works have yet to compare the effectiveness of hierarchical versus non-hierarchical routing protocols in a scenario with a mobile sink, a couple have come close. Chatzigiannakis et al. [3] evaluated a non-hierarchical routing protocol, Directed Diffusion, in scenarios with a mobile sink and a static sink to see how well they compare against each other. This work showed that a moving sink increases the total number of messages collected by up to 140% and consumed 40% less energy than a configuration with a stationary sink.

This work also evaluated Directed Diffusion with a mobile sink in using different movement strategies for the sink. The three movement strategies evaluated were: random sink movement, where the sink moved randomly through the field; biased random sink movement, where the sink tried to move to areas of the field that had yet to report sensor field information; and predetermined movement, where the sink moved in a circular path around the field. This was similar to the work done in [9].

Another work that is related is that done by Guru et al. [5]. In this work,
a hierarchical routing protocol, LEACH, was evaluated under scenarios with a mobile sink and a static sink to see how they compare. This work showed that a sensor field with LEACH was able to collect 15% more data packets per Joule of energy with a mobile base station as opposed to those with a static base station.
Chapter 3

Data-Centric Routing Protocols

In order to evaluate the effectiveness of hierarchical protocols versus non-hierarchical protocols, a better comparison can be provided if all of the protocols were designed for a similar data collection strategy. One such data collection strategy is data-centric routing. In data-centric routing, the sink queries the sensor field for the specific information it is interested in and the nodes respond with only the information requested. Example queries can be a set of nodes (e.g., data from nodes 1, 2, and 3), or a data criteria (e.g., all nodes with a temperature reading greater than 80 degrees). This work assumes a scenario where the sink requests one message from each node.

Three example routing protocols that were designed for a data-centric approach are Flooding, Directed Diffusion, and TEEN. Both Flooding and Directed Diffusion are non-hierarchical routing protocols and TEEN is a hierarchical routing protocol.
3.1 Flooding

Flooding [1] is a non-hierarchical routing protocol and is the least sophisticated of the three routing protocols used in this work. It has no strategy of how to send a message to the sink. When it sends a message, it broadcasts it to all of its neighbors. This continues until the message has reached its destination, or exceeded a maximum number of hops. As this message progresses through the field, each subsequent node broadcasts the message to all its neighbors. This can result in implosion, a situation where a single node contains multiple messages from the same node. Processing duplicate messages means more energy is being spent and more memory is needed for each node.

However, by broadcasting the message to all neighboring nodes, more nodes in the field will have that data at any given point in time. That may present an advantage to a moving sink that moves through the sensor field at a high speed. It may allow more messages to be collected more quickly, thereby reducing the total amount of time that the field needs to send messages.

Figure 3.1 illustrates how messages are transmitted in the Flooding protocol. First, the sink queries for the data it is interested in, then all nodes transmit their data to every one of their neighbors.

3.2 Directed Diffusion

Directed Diffusion [7] is another non-hierarchical, data-centric routing protocol. This protocol uses a more sophisticated routing algorithm than Flooding which attempts to reduce the total number of messages sent in the system. In this protocol, the sink sends out query messages to the field for nodes of interest.
As each node in the field receives the query message, it propagates the message out to its neighboring nodes.

When the query message propagates through the system, each node has a path back to the sink. The set of paths back to the sink are called the gradient field. Next, nodes that have the data that meets the query criteria transmit the data back to the sink. Nodes only transmit their data along one path at a time, but will bias towards choosing the shortest path. In this work, every other transmission is sent through the shortest path and the every other available path is used with equal frequency.

Figure 3.2 shows an example of how Directed Diffusion sends a query message through the sensor field and how that results in transmission paths back to the sink. The first step shows the interest propagation moving through the field. The second step shows the set of all gradient fields back to the sink. The third
Figure 3.2: Illustration of how Directed Diffusion establishes communication routes.

step shows the optimal path back to the sink that is the most frequently used transmission path used by each node.

While Directed Diffusion will reduce the total number of messages in the sensor field in situations with a static sink, a mobile sink may present problems for this protocol. For the time it takes a query message to propagate through the system, the sink may no longer be reachable along those paths.

### 3.2.1 Directed Diffusion Modifications

To address the potential issues associated with only sending messages along one path at a time to reach the sink, a modified version of Directed Diffusion that always sends messages along all available paths was also developed as part of this work. It works exactly the same as traditional Directed Diffusion, except that it transmits messages along all paths back to the sink.

Figure 3.3 shows the modified version of Directed Diffusion. The interest propagation is the same as in traditional Directed Diffusion, but instead of only
Figure 3.3: Illustration of a modified version of Directed Diffusion.

sending messages along one path at a time, the modified Directed Diffusion sends the messages along all available paths simultaneously.

This modification should provide more possibilities to deliver messages to the sink as it moves through the sensor field. However, this modification will also most certainly increase the total number of messages sent through the system in the short term. But if more messages in the short term enables messages to be received by the sink sooner in the long run, then it could be a worthwhile tradeoff.

3.3 TEEN

TEEN [10], or Threshold sensitive Energy Efficient sensor Network protocol, is the only hierarchical routing protocol used in this work. It is also data-centric and uses data querying techniques like Directed Diffusion. The main difference is that nodes in a sensor field using the TEEN protocol are segregated into groups. Each group is designated with a leader node. Each node can only communicate
with other nodes in its group, except for the leader which can communicate with
nodes from other groups.

Figure 3.4 illustrates how TEEN propagation works. In this example, nodes
1, 2, and 3 belong to group 1 with node 2 being the group leader. Nodes 4, 5, 6,
and 7 belong to group 2 with node 4 being the group leader. Interest propagation
moves through the field in the same manner as the other protocols, but as shown
in the second step, the available paths back to the sink are limited to nodes within
the same group, except for leader nodes which can communicate between groups.

Segregating the nodes into hierarchical groups like this aims to increase scala-

bility by reducing the total number of messages that an individual node is capable
of receiving. TEEN experiments showed that in general making 5 percent of the
total nodes in the sensor field into leader nodes produced optimal energy-saving
results [10].

The original TEEN algorithm was designed for the specific use-case of de-
tecting rapid changes in sensed values like a temperature spike. To support this,
the original TEEN experiments only used threshold-sensitive data queries. How-
ever, future experiments made modifications to the routing protocol to query for
specific node data. [11]

3.3.1 TEEN Modifications

While previous TEEN experiments showed that making 5 percent of the nodes
into leaders was optimal, those experiments were conducted using a stationary
sink. A sensor field with a moving sink might show different results. To test that,
this simulator tested TEEN with several different leader percentages.
Figure 3.4: Illustration of the TEEN protocol.
Chapter 4

Sink Movement Strategies

The simulator also supports multiple different sink movement strategies. This allows the simulator to evaluate the effect of sink movement on data collection and message transmission totals, as was done in [3]. The movement strategies supported by the simulator are straight path, random path, biased random path, and predetermined path.

In the cases of all the different movement strategies, the sensor field constructs the sink at the same starting position; diagonally at the top left outside the field.

4.1 Straight

When the sink is configured to run a straight path, the sink simply makes one straight path running directly through the center of the field. This movement strategy will give the sink a very low total time in the field, so a time-efficient routing protocol will be incredibly important for maintaining a high percentage of data collection.
4.1 Straight

When running the straight path, the sensor field will stop collecting data once the sink has passed double the maximum X or Y boundary of the sensor field. So, for example, when the sensor field is 200 units by 250 units, the sensor field will continue collecting data until the sink moves past an X coordinate value of 400, or a Y coordinate value of 500.

4.2 Random

To construct a random path algorithm for the sink, two different approaches were attempted in the sensor field to garner meaningful results while still being random. The initial approach randomly selected a new movement direction for the sink after each iteration. This approach did not work well because the sink did not often move outside of a very small area close to its starting location; it was not a realistic approximation of how a mobile sink might attempt to collect data from a sensor field.

A modification was made to the first attempt at random movement where
the sink would only pick a new direction after every five or ten iterations. This increased the area that the sink would cover, but it often spent time outside the field where it could not collect data from any node. This often produced poor data collection results and did not seem to mimic a realistic strategy that a sink might employ to collect data. A modification to this version was attempted to first move the sink into the center of the field before moving randomly. This decreased the likelihood that the sink would spend time outside the sensor field, but it would not prevent that possibility.

The second and final approach for how the simulator manages random sink movement is to instead of pick a random direction for the sink to travel, the sink picks a random location inside the field to travel. This approach guarantees that the sink spends all of its time inside the sensor field, where the nodes are known to exist, and still move in a random manner. This better approximates a realistic random approach that a sink might employ to collect data.

For this random movement approach, the sink can not randomly select any point in the field. Instead, the set of random points is limited to one of only 64 possible locations. The simulator effectively divides the field into an 8x8 grid of equally sized square blocks and the sink is only allowed to travel to the point at the center of each of those blocks.

The second random sink movement strategy is the only movement strategy used in this work. Also, when using the random path movement strategy, the sensor field simulator stops simulating after two hundred iterations of the sensor field.
4.3 Biased Random

The intent of the biased random movement strategy is to randomly select how the sink moves, but to bias that movement so that the sink attempts to travel to a position in the field where it may prove more successful at collecting messages it has not collected yet.

To move the sink using a biased random approach, the simulator employs the same fundamental base that it uses for the standard random movement strategy: biased random movement selects from one of 64 possible locations to travel to.

The modification made for biased random is that when the sink is picking the next random location to travel to, it examines messages from nodes it has already received. Each message is assumed to contain the position of the node it was generated from and the sink uses that to determine which block that message originated from.

The sink then applies a negative weight to each block depending on how many messages have been received from that node. The sink is more likely to select
a block to visit if it has received fewer messages from that block. The negative weight will never prevent a block from being visited; once 9 messages have been received from a block it will no longer be further penalized. However, if there are blocks in the sensor field that contain very few or no nodes at all, the sink will be more likely to visit those repeatedly over time. That could cause a negative impact to the performance of this movement strategy.

Like the standard random path movement strategy, the sensor field simulator stops simulating after two hundred iterations of the sensor field.

4.4 Predetermined

The predetermined path runs the sink through a pre-planned route through the field, just like how the straight path movement strategy determines its path. In the work done by [9], they found that the most effective predetermined movement strategy they tested was to take a path that traverses the perimeter of the sensor field. The simulator uses the same approach for the predetermined path.

When the simulator runs the sink through the predetermined path, the sink moves around the field in a clockwise rotation, travelling in a rectangular path at the perimeter of the sensor field.

And like both random movement strategies, the simulator stops iterating after two hundred iterations of the sensor field when using the predetermined movement strategy.
Figure 4.3: A sink moving through a sensor field using the predetermined path approach.
Chapter 5

Sensor Field Simulator

To compare the results of hierarchical routing protocols versus non-hierarchical routing protocols in an environment with a mobile sink, a sensor field simulator was developed for this work. The simulator can construct a sensor field and run multiple simulation runs on that field, each with a different variable configuration.

A simulation run consists of a sequence of simulator iterations. Each iteration is a single step in the simulation process where each node generates a new message and/or transmits messages in its queue, and the sink moves.

The simulator supports several variables that can be altered for each simulated run. These variables include:

- sink speed,
- number of nodes in the field,
- node communication radius,
- node memory size,
- message aggregation limit,
• sensor field routing protocol, and

• sink movement strategy

For each simulation, the simulator records the total number of messages sent in order to measure energy efficiency and how many of the nodes queried for were successfully received by the sink in order to measure the completeness of data collection.

5.1 Simulator Assumptions

The simulator operates on several assumptions. First, the nodes are assumed to be randomly distributed within a known geographical area. At the beginning of the simulation, the sink does not know where the nodes are other than somewhere within the pre-defined geographical area. This means that the nodes are not pre-arranged so any sophisticated pre-configured routing algorithms can not be used.

Here is a list of other assumptions that the simulator enforces when constructing the sensor field:

• The sensor field is a connected field; there is a path between each and every node in the field.

• All nodes are constructed within a known grid, and no two nodes can occupy the same space in the grid.

• Nodes are randomly placed, but never move during the course of the simulation.

• Messages never fail to communicate between nodes.
5.2 Simulator Iterations

After constructing the sensor field with the defined variable configuration, the simulator places the sink outside the sensor field and begins stepping through iterations. Each iteration includes the following steps:

1. Process through each node in the field.
   (a) Create a new message for the node and add it to the send queue, if the queue is not full.
   (b) Using the routing protocol of the node, transmit messages from its send queue to its neighbors in the sensor field. The sink may be included as a neighbor, if it is in range and in such cases is a valid target for the routing protocol.

2. Process the sink updates.
   (a) Consult the movement strategy logic of the sink and update the direction of travel of the sink, if needed.
   (b) Move the sink to a new position in the sensor field based on its direction of travel and speed.
   (c) Send out an updated query message to the sensor field if new messages were collected in this iteration.

During each iteration, the sensor field tracks statistics for how many messages the sink received, how many total messages were transmitted by nodes, and which nodes each message visited. These stats are available after the simulator is done running iterations on the field. The simulator stops running after 200 iterations, or if the movement strategy determines that it should stop sooner.
5.3 Routing Protocols

The simulator can be configured to run simulations using each of the routing protocols described in Section 3: Flooding, Directed Diffusion, and TEEN. Both the versions without optimizations and with optimizations are available configuration options.

5.4 Movement Strategies

The simulator also supports multiple different sink movement strategies. This allows the simulator to evaluate the effect of sink movement on data collection and message transmission totals, as was done in [3]. The movement strategies supported by the simulator are straight path, random path, biased random path, and predetermined path.

In the cases of all the different movement strategies, the sensor field constructs the sink at the same starting position; diagonally at the top left outside the field.
Chapter 6

Experiment Methodology

To evaluate the effectiveness of non-hierarchical routing protocols versus hierarchical routing protocols using the simulator, several experiments were run using different sensor field configurations. A sensor field configuration is made up of several different variables, defined below.

Each configuration tested was run on one hundred different sensor fields and the results of those hundred fields was averaged to obtain the overall effectiveness of that configuration. Each configuration was also tested under each routing protocol and movement strategy.

6.1 Simulation Variables

In addition to routing protocol and sink movement strategy, the following variables were configured in the simulations run to evaluate the routing protocols under different circumstances:

- Sink Speed
• Number of Nodes

• Node Communication Radius

Sink speed is the number of units that a sink moves per iteration of the sensor field. When sink speed is not a variable value being measured, the default value of five is used.

Number of nodes is the total number of nodes in the sensor field. When the number of nodes is not a variable being measured, the default value used by the simulator is three hundred.

Node communication radius affects the maximum range that each node and the sink can transmit messages. The larger the communication radius, the more nodes each node can communicate with. When communication radius is not a variable being measured, the simulator uses a default value of twenty five.

6.2 Evaluation Criteria

To evaluate the effectiveness of each configuration, two metrics are considered:

1. The percentage of data left uncollected by the sink.

2. The average number of messages received per node.

The first metric measures how many total nodes the sink fails to receive at least one message from, divided by the total number of nodes in the sensor field to get a percentage. The lower this percentage is, the fewer iterations, and therefore less total time, the sink needs to spend in the sensor field to get a reasonable collection of data. If the sink needs to spend fewer iterations in the sensor field,
then the sensor field will likely have to transmit fewer messages. Fewer messages means less total energy spent by the sensor field.

The second metric measures the total number of messages received by other nodes, normalized by the number of iterations run in the simulation. This metric is effectively a measure of energy expended by the sensor field. Fewer messages received means fewer messages transmitted, which means less energy expended by each node.

Since both metrics indicate better success when their respective values are lower, the product of the two metrics can also be used to evaluate the effectiveness of the configuration with respect to both metrics.
Chapter 7

Results

Before comparing hierarchical versus non-hierarchical, the various movement strategies presented in previous works should be evaluated for each routing protocol. After a preferred movement strategy is determined, Flooding, Directed Diffusion, and TEEN are each evaluated independently using their original design and their modified version to determine which one is superior.

Finally, once the movement strategy and preferred version of each protocol is determined, the combination of the best of each category will be compared with each other to compare how hierarchical and non-hierarchical routing protocols compare against each other.

7.1 Testing Movement Strategy

To evaluate which movement strategy produces the best results, each of the base routing protocols (Directed Diffusion, Flooding, and TEEN) were evaluated for how they perform under configurations of sink speed varying between 5 and
50 units, node communication radius varying between 10 and 55 units, and sensor field node count varying from 100 to 500 nodes. Since sink movement does not significantly affect the number of messages transmitted by the field, only the node data collection metric was compared for this test.

### 7.1.1 Directed Diffusion and Movement

When evaluating different sink movement strategies and Directed Diffusion, the results of varying sink speeds are shown in Figure 7.1. Random sink movement produced the best results, on average leaving only 6.80% of the nodes in the field left uncollected. Biased random sink movement performed second best, leaving 12.88% of nodes uncollected, with predetermined and straight paths leaving 38.60% and 68.33% of nodes uncollected, on average.
Figure 7.2: Evaluation of node collection percentage for Directed Diffusion with varying node communication radii.

Figure 7.2 shows the results of evaluating Directed Diffusion and different node communication radii. In this scenario, random sink movement again produced the best results, leaving only 27.62% of nodes left uncollected. Biased random and random sink movement were relatively close in comparison, leaving behind only 29.92% and 32.34%, respectively. Straight path had the worst performance by a large margin, leaving behind 54.53% of nodes.

The results of evaluating Directed Diffusion paths and sensor field node count is shown in Figure 7.3. All 4 movement strategies performed in the same order, with random, biased random, predetermined, and straight paths resulting in leaving behind an average of 31.68%, 34.75%, 35.47% and 60.42% of nodes.

In all three evaluations of Directed Diffusion, random sink movement performed the best.
7.1.2 Flooding and Movement

The results of testing Flooding movement strategies with variable sink speed is shown in Figure 7.4. Under these conditions, random sink movement performed best, leaving behind only 6.81% of nodes. Biased random, predetermined, and straight sink movement left behind 12.56%, 43.39%, and 90.36% of nodes, respectively.

The results of testing Flooding movement strategies with variable sink communication radii is shown in Figure 7.5. Under these conditions, random sink movement again performed best, leaving behind only 27.62% of nodes. Biased random, predetermined, and straight sink movement left behind 29.92%, 32.34%, and 54.53% of nodes, respectively.

The results of testing Flooding movement strategies with variable sensor field
Figure 7.4: Evaluation of node collection percentage for Flooding for varying sink speed.
Figure 7.5: Evaluation of node collection percentage for Flooding with varying node communication radii.
node counts is shown in Figure 7.6. Under these conditions, random sink movement again performed best, leaving behind only 35.25% of nodes. Biased random, predetermined, and straight sink movement left behind 38.60%, 40.52%, and 66.53% of nodes, respectively.

Like Directed Diffusion, random sink movement performed best under all tests for the Flooding protocol.

7.1.3 TEEN and Movement

The results of testing TEEN movement strategies with variable sink speed is shown in Figure 7.7. Under these conditions, random sink movement performed best, leaving behind only 34.31% of nodes. Biased random, predetermined, and
Figure 7.7: Evaluation of node collection percentage for TEEN with varying sink speed.

straight sink movement left behind 43.94%, 43.81%, and 86.56% of nodes, respectively.

The results of testing TEEN movement strategies with variable sink communication radii is shown in Figure 7.8. Under these conditions, random sink movement again performed best, leaving behind only 34.69% of nodes. Biased random, predetermined, and straight sink movement left behind 39.05%, 37.44%, and 56.23% of nodes, respectively.

The results of testing TEEN movement strategies with variable sensor field node counts is shown in Figure 7.9. Under these conditions, random sink movement again performed best, leaving behind only 31.04% of nodes. Biased random, predetermined, and straight sink movement left behind 38.00%, 33.58%, and 61.55% of nodes, respectively.
Figure 7.8: Evaluation of node collection percentage for TEEN with varying node communication radii.
Figure 7.9: Evaluation of node collection percentage for TEEN with varying sensor field node populations.
7.1.4 Movement Conclusion

When evaluating all 3 tests for all 3 protocols, the results were always the same. Random sink movement always performed the best, with biased random, predetermined, and straight sink movement coming in second, third, and last in all cases.

It is somewhat surprising that random sink movement produced the best collection results, not the biased random or predetermined path strategies. Both the random movement strategies visit a much larger percentage of the sensor field more often than the predetermined path, which likely contributes to their success; this makes it more likely to reduce the number of steps required to reach the sink.

A biased random path would be expected to enhance this effect, and make it more likely to leave a short path between the sink and nodes that have yet to collect data. However, biasing the sink towards areas of the sensor field that have yet to receive data can cause the sink to more often visit areas of the sensor field where no nodes exist. This would reduce the total amount of time that the sink could collect data from the field.

These reasons likely explain why random sink movement produces the best data collection results. The total number of messages transmitted for each protocol did not change significantly when varying the movement strategy, so it was not a factor in the decision of which movement strategy to evaluate use for future tests. Because random sink movement performed best for data collection evaluation, it was used for all further tests in this work.
7.2 Testing Modified Protocols

In 3.2.1 and 3.3.1, modifications to Directed Diffusion and TEEN that might enhance their results were proposed. However, these modifications may or may not actually improve on the performance of the original protocols.

Before being able to compare hierarchical and non-hierarchical protocols, the best configuration for Directed Diffusion and TEEN needed to be decided upon. This section examines the modifications made to Directed Diffusion and TEEN to see if they performed better than the originals.

7.2.1 Directed Diffusion

To potentially improve on Directed Diffusion, a modification was made to transmit data along all discovered paths back to the sink, instead of just one at a time. This would theoretically distribute data across a larger percentage of nodes in the sensor field at any given point in time, thereby increasing the likelihood that the sink would be in range of a node that contains that data. This modification was named Directed Diffusion Send All.

To test Directed Diffusion, the same suite of tests were run that were used to test movement strategies. However, in these tests random sink movement was the only movement strategy used and Directed Diffusion and Directed Diffusion Send All were the only two protocols used. Also, total sensor field message transmission was also relevant, so that was analyzed in addition to data collection.
Directed Diffusion and Sink Speed

The results of testing Directed Diffusion protocols with variable sink movement speed is shown in Figures 7.10, 7.11, and 7.12. Under the data collection evaluation criteria, traditional Directed Diffusion performed the best in all test cases, leaving behind an average of only 6.80% of nodes as opposed to the 13.55% that the Send All version left behind.

Message transmission, however, saw better results for Send All for the speeds tested, averaging 0.20 messages per step on average as opposed to Directed Diffusion which scored 0.30. Message transmission for traditional Directed Diffusion did decrease as speed increased, while Send All remained mostly the same.

And finally, the product of the two metrics showed better results for Directed Diffusion, narrowly edging out Send All with a percentage score of 0.02 to 0.03. Since traditional Directed Diffusion performed better in two out of the three metrics, and also trended better in message transmission as the sink speed increases,
Figure 7.11: Evaluation of Directed Diffusion message transmission with varying sink speed.

it seems better suited for scenarios with varying sink speed.

**Directed Diffusion and Node Communication Radius**

The results of testing Directed Diffusion protocols with variable node communication radii is shown in Figures 7.13, 7.14, and 7.15. Under the data collection evaluation criteria, traditional Directed Diffusion performed the best in all test cases, leaving behind an average of only 27.62% of nodes as opposed to the 31.60% that the Send All version left behind.

Message transmission, however, again saw better results for Send All for the radii tested, averaging 0.18 messages per step on average as opposed to Directed Diffusion which scored 0.30. Message transmission for traditional Directed Diffusion again decreased as radii increased, while Send All increased slightly. Traditional Directed Diffusion began to perform better after the radius was increased.
Figure 7.12: Evaluation of the product of Directed Diffusion data collection and message transmission with varying sink speed.

Figure 7.13: Evaluation of Directed Diffusion data collection with varying node communication radii.
And finally, the product of the two metrics showed better results for Directed Diffusion Send All, narrowly edging out traditional Directed Diffusion with a score of 0.05 to 0.10. Since Directed Diffusion Send All performed better in two out of the three metrics it seems better suited for scenarios with varying node communication radius. However, traditional Directed Diffusion performed better under conditions with larger communication radius, so it would be better suited for fields that are less dense.

**Directed Diffusion and Node Count**

The results of testing Directed Diffusion protocols with variable sensor field node count is shown in Figures 7.16, 7.17, and 7.18. Under the data collection evaluation criteria, traditional Directed Diffusion performed the best in all test
Figure 7.15: Evaluation of the product of Directed Diffusion data collection and message transmission with varying node communication radii.

cases, leaving behind an average of only 31.68% of nodes as opposed to the 36.58% that the Send All version left behind.

Message transmission, however, again saw better results for Send All for the node counts tested, averaging 0.19 messages on average as opposed to Directed Diffusion’s 0.35. Message transmission for traditional Directed Diffusion steadily increased as the node count increased, while Send All slowly decreased.

And finally, the product of the two metrics showed better results for Directed Diffusion Send All, beating traditional Directed Diffusion with a score of 0.07 to 0.12. Since Directed Diffusion Send All performed better in two out of the three metrics it seems better suited for scenarios with varying sensor field node counts.
Figure 7.16: Evaluation of Directed Diffusion data collection with varying sensor field node populations.

Figure 7.17: Evaluation of Directed Diffusion message transmission with varying sensor field node populations.
Figure 7.18: Evaluation of the product of Directed Diffusion data collection and message transmission with varying sensor field node populations.

Directed Diffusion Conclusion

After comparing standard Directed Diffusion with Directed Diffusion Send All, Directed Diffusion yielded a higher data collection percentage in all cases, but in almost all cases the number of messages transmitted through the system was also greater for Directed Diffusion.

To determine which Directed Diffusion protocol performed best overall, the results of each of the product metrics, which is the product of the data collection and message transmission metrics, was evaluated. Those results are shown in Table 7.1. The first three columns show the results of the speed, radius, and node tests. The fourth column, the aggregate column, is the sum of the previous three columns. The lowest score in the aggregate column represents the best overall protocol.
Table 7.1: Results of the product test for all Directed Diffusion modification tests.

<table>
<thead>
<tr>
<th></th>
<th>Speed</th>
<th>Radius</th>
<th>Node</th>
<th>Aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed Diffusion</td>
<td>0.02</td>
<td>0.10</td>
<td>0.12</td>
<td>0.24</td>
</tr>
<tr>
<td>Directed Diffusion Send All</td>
<td>0.03</td>
<td><strong>0.05</strong></td>
<td><strong>0.07</strong></td>
<td><strong>0.15</strong></td>
</tr>
</tbody>
</table>

Evaluating using the aggregate of all product test results, Directed Diffusion Send All produces the best results, coming in with a score of 0.15, which is lower than the score of 0.24 from the traditional Directed Diffusion tests. Since Directed Diffusion Send All performed better, it was chosen to represent Directed Diffusion in the final test of hierarchical vs. non-hierarchical protocols.

7.2.2 TEEN

In the original development of TEEN determined that making 5 percent of the sensor field into group leaders was optimal. However, this was done in a scenario without a moving sink. To potentially improve on TEEN, different group sizes were tested.

To test TEEN, 5 different leader percentages were tested: 5%, 10%, 20%, 40% and 50%. The exact same test suites and evaluation procedures used to test the Directed Diffusion modifications were also used to test TEEN.

TEEN and Sink Speed

The results of testing TEEN protocols with variable sink movement speed is shown in Figures 7.19, 7.20, and 7.21. Under the data collection evaluation criteria, TEEN performed best with 20% of the group defined as leaders. This configuration left only 5.08% of nodes uncollected. Configurations with 10% and
30% came in a close second and third, leaving only 5.11% and 5.96% of nodes uncollected, respectively. 40% and 5% came in last, leaving 16.42% and 34.31% of nodes uncollected respectively.

Message transmission testing showed that message transmission decreases as the percentage of total leaders increases. The results for leader percentages of 5%, 10%, 20%, and 40% were 1.06, 1.02, 0.88, 0.61, and 0.50. Message transmission also stayed consistent despite sink speed, always resulting in the same number of messages transmitted.

And finally, the product of the two metrics came in with a tie for best performance, with 20% and 40% leader percentages producing a score of 0.04. 10% and 50% came in close behind with scores of 0.05 and 0.08 respectively. Leader
Figure 7.20: Evaluation of TEEN message transmission with varying sink speed.
Figure 7.21: Evaluation of the product of TEEN data collection and message transmission with varying sink speeds.

Percentage of 5% came in last by far scoring a high 0.37.

**TEEN and Node Communication Radius**

The results of testing TEEN protocols with variable node communication radii is shown in Figures 7.22, 7.23, and 7.24. Under the data collection evaluation criteria, TEEN performed best with 10% of the group defined as leaders. This configuration left only 16.26% of nodes uncollected. Configurations with 20% and 30% came in a close second and third, leaving only 17.88% and 20.68% of nodes uncollected, respectively. 40% and 5% came in last, leaving 29.14% and 34.69% of nodes uncollected respectively.
Figure 7.22: Evaluation of TEEN data collection with varying node communication radii.
Figure 7.23: Evaluation of TEEN message transmission with different node communication radii.

Message transmission testing again showed that message transmission decreases as the percentage of total leaders increases. The results for leader percentages of 5%, 10%, 20%, and 40% were 1.06, 1.03, 0.88, 0.61, and 0.50. Message transmission stayed generally consistent despite node communication radius, staying nearly the same for all data points.

And finally, the product of the two metrics favored 40% leader percentage which came in with a score of 0.13. 50%, 20% and 10% came in close behind with scores of 0.15, 0.16, and 0.17. Leader percentage of 5% came in last by far scoring a high 0.37.
Figure 7.24: Evaluation of the product of TEEN data collection and message transmission with varying node communication radii.
The results of testing TEEN protocols with variable sensor field node counts is shown in Figures 7.25, 7.26, and 7.27. Under the data collection evaluation criteria, TEEN performed best with 20% of the group defined as leaders. This configuration left only 19.09% of nodes uncollected. Configurations with 10% and 40% came in a close second and third, leaving only 19.39% and 23.61% of nodes uncollected, respectively. 5% and 40% came in last, leaving 31.04% and 36.95% of nodes uncollected respectively.

Message transmission testing again showed that message transmission decreases as the percentage of total leaders increases. The results for leader percentages of 5%, 10%, 20%, and 40% were 1.06, 1.02, 0.88, 0.61, and 0.51. Message
transmission stayed relatively consistent despite the number of nodes, staying nearly the same for all data points.

And finally, the product of the two metrics favored 40% leader percentage which came in with a score of 0.14. 20%, 50% and 10% came in close behind with scores of 0.17, 0.20, and 0.19. Leader percentage of 5% came in last by far scoring a high 0.33.

**TEEN Node Investigation**

TEEN data collection produced oscillating results when testing for varying number of nodes. This test case was the only one that produced oscillating
Figure 7.27: Evaluation of the product of TEEN data collection and message transmission with varying sensor field node populations.
results. To investigate the cause for this, an additional test was run for TEEN to discover why these results varied in this manner.

The TEEN node test checked every node count between 100 and 150. The results are shown in Figure 7.28. As the figure shows, each TEEN configuration experiences jumps at regular intervals. Each configuration significantly changes results whenever the number of nodes becomes high enough to add a new node to the list of leaders.

For example, the 10% leader percentage graph jumps from 17.27% to 24.95% nodes left uncollected for 109 and 110 nodes, respectively. This is the point where the number of leaders increases from 10 to 11. Similarly, the percentage of nodes left uncollected decreases from 27.03% to 16.65% when moving between 119 and 120 nodes, respectively. This is the point where the number of leaders increases from 11 to 12.

For every configuration, the percentage of nodes left uncollected dropped when going from an odd number of nodes to an even number of nodes. Similarly, the percentage always increased when going from an even number of nodes to an odd number of nodes. This suggests that an even number of leaders is always more desirable.

**TEEN Conclusion**

To determine which TEEN configuration performed best overall, the results of each of the product metrics, which is the product of the data collection and message transmission metrics, was evaluated. Those results are shown in Table 7.2. The first three columns show the results of the speed, radius, and node tests. The fourth column, the aggregate column, is the sum of the previous three
The results for TEEN were quite clear, showing that using 40% of the group as leaders produced the best results in all 3 tests. Its aggregate came in with a score of 0.31. 20% of nodes as leaders produced the second best results, producing a score of 0.37. 10%, 50%, and 5% leaders came in at the bottom, producing scores of 0.41, 0.43, and 1.07 respectively.

Since 40% leaders produced the best results, it was used in the final test of
hierarchical vs. non-hierarchical protocols.

7.3 Testing Hierarchical versus Non-Hierarchical

After performing the preliminary tests, the stage was set to run the final tests to evaluate hierarchical and non-hierarchical routing protocols. The movement strategy testing showed that random sink movement produced the best results, so random sink movement was used for all of the final tests.

Similarly, evaluation of Directed Diffusion showed that the Directed Diffusion Send All modification performed better than traditional Directed Diffusion, so the modified version was used for the final tests. And lastly, the TEEN configuration tests showed that using a 40% leader percentage produced optimal results so that configuration was used for the final tests.

7.3.1 Hierarchical vs. Non-Hierarchical and Sink Speed

The results of testing hierarchical and non-hierarchical protocols with variable sink movement speed is shown in Figures 7.29, 7.30, and 7.31. Under the data collection evaluation criteria, TEEN performed the best, leaving behind an average of only 5.96% of nodes on average. Flooding and Directed Diffusion Send All came in behind, leaving 6.81% and 13.55%, respectively. TEEN performed much better at slower speeds, but was surpassed by Flooding as speed increased. Flooding was not better enough at higher speeds as TEEN still performed best on average.

In the message transmission test, Directed Diffusion Send All clearly performed best, averaging only 0.20 messages per step, while TEEN and Flooding
Figure 7.29: Evaluation of hierarchical and non-hierarchical data collection with varying sink speed.

came in second and third with results of 0.61 and 1.15 messages per step, respectively. All protocols were mostly consistent, with very little variation in results as the sink speed changed.

And finally, the product of the two metrics showed better results for Directed Diffusion Send All, coming in with a score of 0.02. TEEN came in second with a score of 0.04 and Flooding came in third with a score of 0.08.

7.3.2 Hierarchical vs. Non-Hierarchical and Node Communication Radius

The results of testing hierarchical and non-hierarchical protocols with variable node communication radii is shown in Figures 7.32, 7.33, and 7.34. Under the data collection evaluation criteria, TEEN again performed the best, leaving
behind an average of only 20.68% of nodes on average. Flooding and Directed Diffusion Send All came in second and third, leaving behind 29.94% and 31.60% respectively. In this scenario, the protocols were very consistent with TEEN producing the best results and Directed Diffusion producing the worst results for almost every individual data point.

In the message transmission test, Directed Diffusion Send All clearly performed best, averaging only 0.05 messages per step, while TEEN and Flooding came in second and third with results of 0.61 and 1.15 messages per step, respectively. All protocols were mostly consistent, with very little variation in results as the node communication radius changed. These averages were exactly the same as the tests for sink speed.

And finally, the product of the two metrics again showed better results for
Figure 7.31: Evaluation of the product of Hierarchical vs. Non-Hierarchical data collection and message transmission with varying sink speed.

Figure 7.32: Evaluation of hierarchical and non-hierarchical data collection with varying node transmission radii.
Figure 7.33: Evaluation of hierarchical and non-hierarchical message transmission with varying node communication radii.

Directed Diffusion Send All, coming in with a score of 0.05. TEEN again came in second with a score of 0.13 and Flooding came in third with a score of 0.35.

7.3.3 Hierarchical vs. Non-Hierarchical and Node Count

The results of testing hierarchical and non-hierarchical protocols with variable node communication radii is shown in Figures 7.35, 7.36, and 7.37. Under the data collection evaluation criteria, TEEN again performed the best, leaving behind an average of only 23.61% of nodes on average. Flooding and Directed Diffusion Send All came in second and third, leaving behind 35.25% and 36.58% respectively. In this scenario, TEEN was consistently the best, and was the only protocol to perform better as the number of nodes increased.

In the message transmission test, Directed Diffusion Send All again performed
Figure 7.34: Evaluation of the product of Hierarchical vs. Non-Hierarchical data collection and message transmission with varying node communication radii.

Figure 7.35: Evaluation of hierarchical and non-hierarchical data collection with varying sensor field node populations.
Figure 7.36: Evaluation of hierarchical and non-hierarchical message transmission with varying sensor field node populations.

best, averaging only 0.19 messages per step, while TEEN and Flooding came in second and third with results of 0.61 and 1.15 messages per step, respectively. TEEN and Flooding did stay consistent as the node count increased, but Directed Diffusion Send All got slightly better as the node count increased.

And finally, the product of the two metrics again showed better results for Directed Diffusion Send All, coming in with a score of 0.07. TEEN came in second with a score of 0.14 and Flooding came in a distant third with a score of 0.41.

7.3.4 Hierarchical vs. Non-Hierarchical Conclusion

To determine which protocol performed best overall in the final tests, the results of each of the product metrics, which is the product of the data collection
Figure 7.37: Evaluation of the product of Hierarchical vs. Non-Hierarchical data collection and message transmission with varying sensor field node populations.
Table 7.3: Results of the product test for all hierarchical vs. non-hierarchical tests.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Speed</th>
<th>Radius</th>
<th>Node</th>
<th>Aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed Diffusion Send All</td>
<td>0.03</td>
<td>0.05</td>
<td>0.07</td>
<td>0.15</td>
</tr>
<tr>
<td>Flooding</td>
<td>0.04</td>
<td>0.35</td>
<td>0.41</td>
<td>0.80</td>
</tr>
<tr>
<td>TEEN</td>
<td>0.08</td>
<td>0.13</td>
<td>0.14</td>
<td>0.35</td>
</tr>
</tbody>
</table>

and message transmission metrics, were evaluated. Those results are shown in Table 7.3. The first three columns show the results of the speed, radius, and node tests. The fourth column, the aggregate column, is the sum of the previous three columns. The lowest score in the aggregate column represents the best overall protocol.

In all three tests, Directed Diffusion Send All performed the best, aggregating to a total score of 0.15. TEEN came in second scoring 0.35. Flooding came in a distant third, scoring 0.80.

From these results, Directed Diffusion Send All, a non-hierarchical routing protocol, has shown that it is the best general routing protocol for varying sink speed, node communication radius, and sensor field population count.
Chapter 8

Future Work

This work evaluated the effectiveness of hierarchical and non-hierarchical routing protocols in a scenario with a moving sink. Another scenario that could be tested to further evaluate hierarchical and non-hierarchical protocols would be to test with multiple mobile sinks, as was done in [4]. Having multiple sinks in the sensor field might favor Directed Diffusion the least since it only sends messages in one direction at a time. Sending messages in multiple directions, like TEEN and Flooding do, could result in fewer messages left uncollected.

Another scenario that might produce different results for testing hierarchical and non-hierarchical routing protocols would be to periodically stop the sink when it reaches a location that yields a greater data collection rate. This is similar to the work done in [16]. Keeping the sink stationary within range of multiple TEEN groups at the same time might yield a higher data collection rate than Directed Diffusion and stopping in those areas may prove beneficial.

Additionally, other metrics could be used to evaluate these three protocols. This work evaluates total energy expenditure, but total network lifetime is not
examined. The first node to consume all its energy under Directed Diffusion might be significantly better or worse than TEEN or Flooding. For sensor fields like ZebraNet where nodes need to conserve energy for long periods of time and the data from each node is important, total network lifetime might be more critical than average energy consumption across the field.
Chapter 9

Conclusion

In this work, three routing protocols were evaluated under a scenario with a moving sink: Directed Diffusion, Flooding, and TEEN. All are data-centric routing protocols, but Directed Diffusion and Flooding are non-hierarchical routing protocols while TEEN is a hierarchical protocol. All three of these protocols were developed under the assumption of a sensor field with a static sink.

Before comparing these protocols against each other, several other configurations were first tested. Different sink movement strategies were evaluated against all three protocols to determine which one produced the best result. Also, modifications to Directed Diffusion and TEEN were evaluated to see if they improved upon the original configurations.

The movement strategy evaluation showed that random sink movement performed best, averaging 26.20% of nodes left behind in the data collection evaluation. Biased random sink movement, predetermined path, and straight path produced average results of 31.07%, 37.50%, and 66.56% respectively. Because of this result, random sink movement was used for all further evaluations.
The Directed Diffusion modification tested to see if sending messages along all paths back to the sink produced better results than only sending along only one path at a time. The results were close, but the modified Directed Diffusion Send All protocol performed slightly better, producing an aggregate score of 0.15 versus traditional Directed Diffusion which produced a score of 0.24. Because of this result, the modified Directed Diffusion was used for the final evaluation.

TEEN modification testing tested what percentage of leaders produced optimal results. Previous works that used a stationary sink showed that 5% leader percentage was best, but with a moving sink 40% leader percentage produced the best results with an aggregate score of 0.31. The order of the other percentages tested was: 30%, 10%, 50%, and 5%, with aggregate scores of 0.37, 0.41, 0.43, and 1.07. Because of this result, TEEN with 40% leaders was used for the final evaluation.

In the final evaluation, Directed Diffusion Send All, a non-hierarchical protocol, performed best on average, showing a score of 0.15 on the aggregation of the product metrics of the three tests. TEEN came in second, scoring 0.35, while Flooding came in a distant third, scoring 0.80.

While best on average, the Directed Diffusion Send All was not clearly the best protocol in all categories. For all tests of data collection percentage TEEN always produced the best results, but at the cost of slightly higher message transmission than Directed Diffusion. This means TEEN would be a better choice for a problem space where data collection rate was more important than energy consumption.

Directed Diffusion Send All always transmitted the fewest messages in these tests, however, and is clearly the best choice if conservation of energy is more
important than collecting all the data from the sensor field.
Bibliography


