BATSEN: Modifying the BATMAN Routing Protocol for Wireless Sensor Networks

Nelson Henry Powell III
nhp8080@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
BATSEN: Modifying the BATMAN Routing Protocol for Wireless Sensor Networks

By:
Nelson Henry Powell III

Committee Members:
Dr. Sumita Mishra
Dr. Yin Pan
Dr. Andres Kwasinski

In partial fulfillment of the requirements for the degree of
Master of Science in Computing Security

Rochester Institute of Technology
B. Thomas Golisano College of Computing & Information Sciences
Department of Computing Security

May 8th, 2018
Acknowledgments

I’d like to take this opportunity to thank my wife, my love, Emilia; for putting up with the deployments, active duty, and working two jobs, all while going back to school to complete this degree. None of this would have been possible without your love and support. You are my motivation for being a better me. To my children, Connor and Iliana, for teaching me to look at the world as if it were for the first time; and for all the snuggles, hugs, and kisses that happily kept me from my school work. You are both my favorite and most important accomplishments in my life. May you both achieve more than me and maintain the fresh outlook on life.

I’d like to thank my parents for everything. You’ve both been my number one fans all my life. I wouldn’t be at this point in my life without your guidance, wisdom, and love. Dad, your candor, spirituality, and introspection has giving me the tools to strive to be the best man I can be and learn from my mistakes. I most definitely inherited your curiosity and engineering skills. Mom, thank you for all your nurturing and affection throughout my life, for being my sounding board, and being the driving force in my education. And thank you for coming out of retirement and giving that final critical review.

I’d like to thank my co-workers, Dr. Michael Kurdziel for all your support and assistance reviewing my thesis, David Weiss for being the first victim to review my paper and provide valuable feedback, and James Hood for helping me with a last minute save that really pulled the paper together.

To Dr. Kwasinski, I’ve enjoyed our discussions over the years, both regarding this thesis as well as other research topics. I appreciate all the help you’ve provided throughout this project and look forward to working with you again.

Finally, I’d like to thank Dr. Sumita Mishra. You never gave up on me and ensured I was able to achieve my goals. For your insight and guidance throughout this endeavor; and most of all, for your empathy regarding the challenges of balancing work, family, and education. I admire all that you have accomplished and you gave me the reality check of how attainable this thesis is.
Abstract

The proliferation of autonomous Wireless Sensor Networks (WSN) has spawned research seeking power efficient communications to improve the lifetime of sensor motes. WSNs are characterized by their power limitations, wireless transceivers, and the converge-cast communications techniques. WSN motes use low-power, lossy radio systems deployed in dense, random topologies, working sympathetically to sense and notify a sink node of the detectable information. In an effort to extend the life of battery powered motes, and hence the life of the network, various routing protocols have been suggested in an effort to optimize converge-cast delivery of sensor data. It is well known that reducing the overhead required to perform converge-cast routing and communications reduces the effects of the primary power drain in the mote, the transceiver. Furthermore, WSNs are not well protected; network security costs energy both in computation and in RF transmission. This paper investigates the use of a Mobile Ad-hoc Networking (MANET) routing protocol known as B.A.T.M.A.N. in WSN. This thesis proposes that the features of B.A.T.M.A.N. in the MANET realm may prove beneficial to the WSN routing domain; and that slight modifications to the routing technique may prove beneficial beyond current protocol technologies. The B.A.T.M.A.N. variant will be compared against the contemporary LEACH WSN routing protocol to discern any potential energy savings.
Table of Contents

1 INTRODUCTION ............................................................................................................. 1
  1.1 PROPOSED SOLUTION ............................................................................................... 2
  1.2 PURPOSE .................................................................................................................... 3

2 BATMAN ROUTING PROTOCOL .................................................................................. 4
  2.1 PROTOCOL DESIGN ................................................................................................ 5
  2.2 SIMULATING BATMAN ............................................................................................. 7

3 TEST CASES ................................................................................................................ 11

4 USING NS-3 TO SIMULATE SENSORS ...................................................................... 24
  4.1 PHYSICAL RF MODEL REUSE ................................................................................. 24
  4.2 SENSOR MAC MODULE ......................................................................................... 29
  4.3 SENSOR HELPER CLASS ......................................................................................... 31
  4.4 POWER DISSIPATION ............................................................................................. 32

5 THE LEACH PROTOCOL ............................................................................................. 33
  5.1 BENEFITS OF LEACH .............................................................................................. 35
  5.2 CONVERTING NS-2 BASELINE CODE .................................................................... 36
    5.2.1 Assumption 1 ........................................................................................................ 37
    5.2.2 Assumption 2 ........................................................................................................ 38
    5.2.3 Assumption 3 ........................................................................................................ 38
    5.2.4 Assumption 4 ........................................................................................................ 39
    5.2.5 Assumption 5 ........................................................................................................ 41
    5.2.6 Assumption 6 ........................................................................................................ 42
    5.2.7 Assumption 7 ........................................................................................................ 43
    5.2.8 Assumption 8 ........................................................................................................ 43
    5.2.9 Assumption 9 ........................................................................................................ 45
  5.3 A LEACH PROTOCOL SPECIFICATION ................................................................. 45
    5.3.1 State Machine Description ................................................................................ 46
      5.3.1.1 Initialization State ......................................................................................... 47
      5.3.1.2 Advertisement State .................................................................................... 47
      5.3.1.3 Cluster Setup Phase ..................................................................................... 49
      5.3.1.4 Schedule Creation Phase ............................................................................. 50
      5.3.1.5 Data Transmission Phase .......................................................................... 51
    5.3.2 Protocol Timing Analysis .................................................................................. 52
  5.4 LEACH RESULTS ..................................................................................................... 58
    5.4.1 Network Death Rate ......................................................................................... 59
    5.4.2 Number of Cluster Heads per Round ................................................................. 63
    5.4.3 LEACH Packet Rate per Round ....................................................................... 68
  5.5 THE REAL LEACH .................................................................................................. 70

6 THE BATMAN-SENSOR PROTOCOL ........................................................................... 73
  6.1 LEACH AS A STARTING POINT ............................................................................... 73
    6.1.1 LEACH Protocol Requirements .................................................................... 73

Nelson Henry Powell III
6.1.2  LEACH Environment Assumptions................................................................. 75
6.2   BATMAN INHERITANCE .............................................................................. 77
6.3   BATSEN FRAME FORMATS ......................................................................... 80
6.4   THE BATSEN FINITE STATE MACHINE ......................................................... 84
  6.4.1  Startup Phase ......................................................................................... 85
  6.4.2  Power Collection Phase ........................................................................... 88
  6.4.3  Router Selection Phase ............................................................................ 90
  6.4.4  Data Exchange Phase ............................................................................. 94
  6.4.5  Late Net Entry Operations ..................................................................... 96
6.5   PROTOCOL TIMING ANALYSIS .................................................................... 97

7  RESULTS ........................................................................................................... 99

8  CONCLUSION .................................................................................................... 109
  8.1  KEY CONTRIBUTIONS OF THIS WORK ..................................................... 109
  8.2  SUMMARY OF FINDINGS .......................................................................... 109
  8.3  FUTURE WORK .......................................................................................... 111

9  BIBLIOGRAPHY ............................................................................................... 115
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>NS-3 Software Architecture</td>
<td>8</td>
</tr>
<tr>
<td>2-2</td>
<td>NS-3 Component Diagram</td>
<td>10</td>
</tr>
<tr>
<td>2-3</td>
<td>Three Node Basic Scenario</td>
<td>11</td>
</tr>
<tr>
<td>2-4</td>
<td>Three Node Basic Scenario Mid-way Points</td>
<td>12</td>
</tr>
<tr>
<td>2-5</td>
<td>Three Node Basic Scenario Data and Routing Throughput</td>
<td>13</td>
</tr>
<tr>
<td>2-6</td>
<td>Three Node Basic Scenario Throughput Quad-Plots</td>
<td>16</td>
</tr>
<tr>
<td>2-7</td>
<td>Three Node-High User Traffic Results</td>
<td>17</td>
</tr>
<tr>
<td>2-8</td>
<td>Three Node-High User Traffic Quad-Graph</td>
<td>19</td>
</tr>
<tr>
<td>2-9</td>
<td>Multi-Node Mobility Test Scenario</td>
<td>20</td>
</tr>
<tr>
<td>2-10</td>
<td>Multi-Node Mobility Results</td>
<td>21</td>
</tr>
<tr>
<td>2-11</td>
<td>Multi-Node Mobility Quad Graphs</td>
<td>23</td>
</tr>
<tr>
<td>3-1</td>
<td>NS-3 LR-WPAN PHY Class Structures</td>
<td>25</td>
</tr>
<tr>
<td>3-2</td>
<td>Simulation Boundaries with Respect to Transmitter Radius</td>
<td>28</td>
</tr>
<tr>
<td>3-3</td>
<td>Packet Success Rate per Power Level</td>
<td>29</td>
</tr>
<tr>
<td>3-4</td>
<td>NS-3 Sensor Module and Protocol Inheritance</td>
<td>30</td>
</tr>
<tr>
<td>3-5</td>
<td>NS-3 SensorHelper Class</td>
<td>31</td>
</tr>
<tr>
<td>4-1</td>
<td>Original LEACH States</td>
<td>34</td>
</tr>
<tr>
<td>4-2</td>
<td>Network Compression when R = C</td>
<td>39</td>
</tr>
<tr>
<td>4-3</td>
<td>Network Dilation when R = a</td>
<td>40</td>
</tr>
<tr>
<td>4-4</td>
<td>Network Test Optimization when R = d</td>
<td>41</td>
</tr>
<tr>
<td>4-5</td>
<td>Probability of Node’s Self-Election per Round</td>
<td>44</td>
</tr>
<tr>
<td>4-6</td>
<td>LEACH Finite State Machine</td>
<td>46</td>
</tr>
<tr>
<td>4-7</td>
<td>LEACH Protocol Event Timeline</td>
<td>57</td>
</tr>
<tr>
<td>4-8</td>
<td>Simulation Layout, (A) Bottom Random (B) Bottom Uniform (C) Center Random (D)</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Center Uniform</td>
<td></td>
</tr>
<tr>
<td>4-9</td>
<td>LEACH Lifetime per Cluster Head count, Sink at Center, Uniform Distribution of Nodes</td>
<td>60</td>
</tr>
<tr>
<td>4-10</td>
<td>LEACH Lifetime per Cluster Head count, Sink at Center, Random Distribution of Nodes</td>
<td>61</td>
</tr>
<tr>
<td>4-11</td>
<td>LEACH Lifetime per Cluster Head count, Sink at Bottom, Uniform Distribution of Nodes</td>
<td>62</td>
</tr>
<tr>
<td>4-12</td>
<td>LEACH Lifetime per Cluster Head count, Sink at Bottom, Random Distribution of Nodes</td>
<td>63</td>
</tr>
<tr>
<td>4-13</td>
<td>LEACH Cluster Heads per Round, Sink at Center, Uniform Distribution of Nodes</td>
<td>64</td>
</tr>
<tr>
<td>4-14</td>
<td>LEACH Cluster Heads per Round, Sink at Center, Random Distribution of Nodes</td>
<td>66</td>
</tr>
<tr>
<td>4-15</td>
<td>LEACH Cluster Heads per Round, Sink at Bottom, Uniform Distribution of Nodes</td>
<td>67</td>
</tr>
<tr>
<td>4-16</td>
<td>LEACH Cluster Heads per Round, Sink at Bottom, Random Distribution of Nodes</td>
<td>67</td>
</tr>
</tbody>
</table>
Figure 4-17: LEACH Packet Rate per Round per CH Count, Sink at Center, Uniform Distribution of Nodes

Figure 4-18: LEACH Packet Rate per Round per CH Count, Sink at Center, Random Distribution of Nodes

Figure 4-19: LEACH Packet Rate per Round per CH Count, Sink at Bottom, Uniform Distribution of Nodes

Figure 4-20: LEACH Packet Rate per Round per CH Count, Sink at Bottom, Random Distribution of Nodes

Figure 5-1: BATSEN Frame Format

Figure 5-2: BATSEN Super Frame Format

Figure 5-3: BATSEN Base OGM Frame Format

Figure 5-4: BATSEN RX POWER OGM Frame Format

Figure 5-5: BATSEN Router Selection OGM Frame Format

Figure 5-6: BATSEN Data Frame Format

Figure 5-7: BATSEN Forwarder Frame Format

Figure 5-8: BATSEN Finite State Machine

Figure 5-9: Sink Power Level Evaluation Scenarios

Figure 5-10: BATSEN Protocol Timeline

Figure 6-1: BATSEN vs LEACH, Network Lifetime for all Scenarios

Figure 6-2: BATSEN vs LEACH, Round Length in Seconds

Figure 6-3: Network Lifetime BATSEN vs LEACH, Sink at Center of Unit Circle

Figure 6-4: Network Lifetime BATSEN vs LEACH, Sink at Bottom of Unit Circle

Figure 6-5: BATSEN vs LEACH, Cluster Heads per Round, Sink at Center of Unit Circle

Figure 6-6: BATSEN vs LEACH, Cluster Heads per Round, Sink at Bottom of Unit Circle

Figure 6-7: BATSEN vs LEACH, Direct Connect Nodes per Round, Sink at Center

Figure 6-8: BATSEN vs LEACH, Direct Connect Nodes per Round, Sink at Bottom

Figure 6-9: BATSEN vs LEACH, Packet Rate per Round

Nelson Henry Powell III
Table of Tables

Table 5-1: BATSEN OGM Types ........................................................................................................... 80

Table of Equations

Equation 1 - SINR Calculation based on Signal Power (mW) ......................................................... 28
Equation 2 - LEACH Cluster Head Selection Threshold ................................................................. 48
Equation 3 – Maximum Number of Nodes in a Data Frame Payload ........................................... 52
Equation 4 – Advertisement Phase Timeout Calculations ............................................................... 53
Equation 5 – Scheduling Phase Timeout Calculations ..................................................................... 53
Equation 6 – Data Phase Timeout Calculations .............................................................................. 54
Equation 7 – Sink Data Phase Timeout Calculations ...................................................................... 55
Equation 8 – Null Time-Period ..................................................................................................... 86
Equation 9 – RX Power Time-Period ............................................................................................ 89
Equation 10- Node Scoring Calculation ......................................................................................... 90
Equation 11- Aggregated Node Count Calculation ................................................................. 90
Equation 12- Power to Sink Score Calculation ......................................................................... 92
Equation 13- Router Selection Time-Period .............................................................................. 93
Equation 14- Sink Combine Timeout Calculation ..................................................................... 94
Equation 15- Data Exchange Time-Period .................................................................................... 95
# Abbreviations and Symbols

<table>
<thead>
<tr>
<th>Abbreviation or Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-PPS</td>
<td>1 Pulse Per Second</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>BATMAN</td>
<td>Better Approach To MANET</td>
</tr>
<tr>
<td>BATSEN</td>
<td>BATMAN for Sensor Networks</td>
</tr>
<tr>
<td>bps</td>
<td>Bits per second</td>
</tr>
<tr>
<td>Bps</td>
<td>Bytes per second</td>
</tr>
<tr>
<td>CCA</td>
<td>Clear Channel Assessment</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code-Division Multiple Access</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster Head</td>
</tr>
<tr>
<td>CKS</td>
<td>Checksum</td>
</tr>
<tr>
<td>CLI</td>
<td>Client Nodes</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>COMSEC</td>
<td>Communications Security</td>
</tr>
<tr>
<td>CSMA</td>
<td>Carrier-Sense Multiple Access</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier-Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>dBm</td>
<td>Decibel-milliwatts</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Connection</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>ED</td>
<td>Energy Detection</td>
</tr>
<tr>
<td>ETX</td>
<td>Expected Transmission Count</td>
</tr>
<tr>
<td>FND</td>
<td>First Node Death</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Machine</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HC</td>
<td>Hop Count</td>
</tr>
<tr>
<td>HNA</td>
<td>Host Network Association</td>
</tr>
<tr>
<td>HPP</td>
<td>High Power Peer</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronic Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>International Engineering Task Force</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPv4</td>
<td>Internet Protocol version 4</td>
</tr>
<tr>
<td>LEACH</td>
<td>Low-Energy Adaptive Clustering Hierarchy</td>
</tr>
<tr>
<td>LNE</td>
<td>Late Net Entry</td>
</tr>
<tr>
<td>LPP</td>
<td>Low Power Peer</td>
</tr>
<tr>
<td>LR-Wpan</td>
<td>Low-Rate Wireless Personal Area Network</td>
</tr>
<tr>
<td>LQI</td>
<td>Link Quality Indicator</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Adhoc Network</td>
</tr>
<tr>
<td>MCPS</td>
<td>MAC Data Services</td>
</tr>
<tr>
<td>MLME</td>
<td>MAC Management Services</td>
</tr>
<tr>
<td>MPP</td>
<td>Medium Power Peer</td>
</tr>
<tr>
<td>MPR</td>
<td>Multi-Point Relay</td>
</tr>
<tr>
<td>Abbreviation or Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>MTE</td>
<td>Minimum Transmit Energy</td>
</tr>
<tr>
<td>NoF</td>
<td>Number of Forwarders</td>
</tr>
<tr>
<td>NoN</td>
<td>Number of Nodes</td>
</tr>
<tr>
<td>NTDR</td>
<td>Near Term Digital Radio</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>NS-2</td>
<td>Network Simulator (2\textsuperscript{nd} Generation)</td>
</tr>
<tr>
<td>NS-3</td>
<td>Network Simulator (3\textsuperscript{rd} Generation)</td>
</tr>
<tr>
<td>OGM</td>
<td>Originator Message</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimized Link State Routing</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection Model</td>
</tr>
<tr>
<td>PER</td>
<td>Packet Error Rate</td>
</tr>
<tr>
<td>PD</td>
<td>PHY Data Services</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer (RF) Hardware</td>
</tr>
<tr>
<td>PLME</td>
<td>PHY Management Services</td>
</tr>
<tr>
<td>PRNG</td>
<td>Pseudo Random Number Generator</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectrum Density</td>
</tr>
<tr>
<td>PTP</td>
<td>Point-to-Point</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>ROLL</td>
<td>Routing Over Low-power and Loss-networks</td>
</tr>
<tr>
<td>RSSI</td>
<td>Receive Signal Strength Indicator</td>
</tr>
<tr>
<td>SAP</td>
<td>Service Access Points</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal to Interference and Noise Ratio</td>
</tr>
<tr>
<td>SN</td>
<td>Sequence Number</td>
</tr>
<tr>
<td>SRV</td>
<td>Server Nodes</td>
</tr>
<tr>
<td>ToD</td>
<td>Time of Day</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TLV</td>
<td>Type, Length, Vector</td>
</tr>
<tr>
<td>TQ</td>
<td>Transmission Quality</td>
</tr>
<tr>
<td>TRANSEC</td>
<td>Transmission Security</td>
</tr>
<tr>
<td>TTL</td>
<td>Time To Live</td>
</tr>
<tr>
<td>TX</td>
<td>Transmit</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>WPAN</td>
<td>Wireless Personal Area Network</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
1 Introduction

Wireless Sensor Networks (WSN) exist in a multitude of locations: military, infrastructure, industrial, medical, and emergency response, to name a few. As the wireless sensor technology advances, WSN permeate diverse new and remote environments, relying more and more on limited battery power. Unfortunately, it is the wireless component of a sensor mote which accounts for a majority of the power expenditure. Maximum energy depletion occurs during transmission, actively listening, or receiving on a Radio Frequency (RF) channel. To maintain the life of a sensor mote, it is imperative to reduce the RF activity yet accomplish the task of the sensor.

With the advent of the modern hacker, WSN lifetime may soon take a back seat to the security of the network. Communications Security (COMSEC) and Transmission Security (TRANSEC) costs energy. An increase in security primitives, computation, and complexity directly correlates to an increase in energy expenditure. It stands to reason that a reduction in power consumption due to routing and data delivery optimization results in more power available for the application of network security.

The International Engineering Task Force (IETF) chartered the Routing Over Low-power and Lossy Networks (ROLL) working group to define the requirements and develop corresponding routing protocol standards. This group was proposed in 2007 and chartered in 2008 [1]. Thus far, the ROLL work group published works defining routing and device requirements for each various deployment environments [2] [3] [4] [5]. Though not considered comprehensive, the papers attempt to identify the majority of the common operating environments, terminology, and requirements for WSN per each given scenario. Additional Requests for Comments (RFCs) followed shortly thereafter to develop additional requirements or supporting metrics for WSN routing protocols.

Academics have proposed various routing protocols, commonly categorized as Flat, Hierarchical, or Location Based routing [6]. One of the more popular algorithms is the LEACH protocol [7]. LEACH has spawned a number of variations to compensate for inadequacies found in the original protocol. LEACH is a Hierarchical routing protocol that attempts to evenly distribute the power
consumption across the entire WSN. LEACH claims that the hierarchical routing protocol is superior to flat routing protocols.

In spite of the proliferation of LEACH and other Cluster Head (CH) based approaches, others still continue to investigate Flat routing protocols [8] [9] [10]. Flat routing and hybrid protocols have been shown to compensate or outperform Hierarchical routing protocols in various deployment topologies and for varying data-source transmission rates. Regardless of the academic’s preference for protocol categorization, the ultimate goal of routing in WSN is to reduce transmission overhead and inversely increase the WSN lifetime, all while satisfying the particular requirements of the system.

1.1 Proposed Solution
This thesis explores the use of the B.A.T.M.A.N. protocol, a Flat routing algorithm originally designed for optimized Mobile Adhoc NETworking (MANET) wireless networks, in WSN [11]. BATMAN is a flat decentralized routing protocol; wherein individual nodes have no knowledge of the route to an end-point other than the best next-hop address. BATMAN uses a small User Datagram Protocol (UDP) over Internet Protocol (IP) datagram, with a maximum length of 52 bytes, which is relatively close to the IEEE 802.15.4 frame payload length. BATMAN can operate at either the Network Layer or the Link Layer depending on implementation methodology.

The attractiveness of BATMAN derives from its fast convergence, stable horizontal handover, and low overhead [12]. In BATMAN, each node periodically broadcasts an Originator’s Message (OGM) that includes Source Address, Transmitter’s Address, Time To Live, Hop Count, a Transmission Quality (TQ) metric, and Sequence Number. Receiving nodes retransmit the OGM substituting its address for the Transmitter’s Address in the forwarded OGM and updating the TQ metric with respect to the forwarder. It is up to each node to determine which next hop to use when sending traffic to the source of the OGM. Additionally, by rebroadcasting the OGM, link symmetry is established.

As with any other routing protocol, BATMAN is not without its faults. BATMAN does suffer from reduced throughputs due to temporary node absence [12]. In some scenarios, BATMAN has
shown slightly higher delay or lower bit rates when the source or destination of a packet is in motion [13]. In spite of these faults, there is still a potential use of BATMAN or a BATMAN derivative protocol in WSN.

Based on the fundamentals of WSN, some of the undesirable effects of BATMAN can be nullified. Given the assumption that WSN nodes do not move post deployment [7], the source-destination movement issue is minimized. Furthermore, many of the WSN Media Access Control (MAC) protocols allow periodic synchronization, reducing the impact of temporary hidden nodes [14]. Therefore, node loss will be more likely due to a damaged node or a node suffering a complete power loss.

The proposed solution is to port and modify the BATMAN protocol in support of WSN operating on IEEE 802.15.4 capable sensor motes. Using the fundamentals of BATMAN, the protocol will be enhanced by modifying the basic OGM format; wherein the Sink node generates a periodic synchronization OGM that is forwarded throughout the network. Sensors nodes send their own OGM, but do not forward non-Sink node OGMs. Additionally, the packet format will include, at a minimum, mote residual power level and transmitting power level. These modifications will help establish bi-directional relationships as well as aggregation points in the flat network emulating CHS normally seen in Hierarchical Routing Protocols.

1.2 Purpose
This research compares and contrasts an innovative flat routing protocol, based on the B.A.T.M.A.N. project, against traditional wireless sensor networks routing protocols to effect power efficient, reliable communications that support more realistic deployment environments and operational scenarios, and to return energy for network security.
2 BATMAN Routing Protocol

The BATMAN routing protocol is a derivative work from the IETF OLSR effort. OLSR derived partially from the LEACH protocol, which in turn, was based on the EPLRS radio system. OLSR derives its technical advantage from the use of elected Multi-Point Relays (MPR). MPRs function in a similar manner to that of the CHs in LEACH, except that MPRs support inter cluster routing within Mesh Networks while CHs support Node to Sink forwarding in Sensor Networks.

With OLSR, MPRs reduce the overall broadcast traffic by funneling all broadcast, multicast, and control traffic through nodes elected by their peers to act as local MPRs. However, early testing of OLSR(v1) [15] showed significant problems. Nodes typically selected MPRs with a maximum range in order to reduce the number MPRs elected, but this resulted in unstable and asymmetric links to MPRs. Furthermore, OLSR frequently resulted in routing loops and route flapping, so typical of the MANET environment. Minimum hop-count was the basis of OLSRs routing decision which caused an increase in packet loss due to unstable links and increased collisions.

In early 2004, a group of OLSR designers discovered many of the afore-mentioned deficiencies [16]. As mesh sizes increased, the fundamental features of OLSR became the hindrance of the network. The group of developers evolved their version of OLSR, neutering the protocol of many of the features, in order to achieve a working solution. In the end, removal of the features allowed for some level of functionality, but with no discernable advantage over traditional routing protocols.

It was this group that then decided to pursue a non-OLSR approach to MANET routing in an attempt to alleviate a number of design flaws in OLSR. The new approach centered on the use of link quality rather than hop-counts for route selection, the elimination of MPRs, scaled transmission of control messages based on neighbor hop distance, and limiting Djikstra calculations to local neighborhoods to avoid massive full MESH recompiling. It was this work that became the forbearer to the Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N.). BATMAN was developed as a standard Layer 3 routing protocol. At the time of this writing, the BATMAN algorithm reached revision IV. Additionally, BATMAN version 0.3.5 was frozen, and

Nelson Henry Powell III
the algorithm transitioned to a Layer 2 virtual device driver exchanging frames rather than IP packets.

2.1 Protocol Design
The BATMAN protocol identifies participating nodes as Originators; all originators periodically transmit Originator Messages (OGM), announcing their presence and knowledge of distant neighbors. BATMAN is a proactive routing protocol, but BATMAN does not transfer and converge complete routing tables amongst peers. BATMAN attempts to determine the best bi-directional neighbor in which to send a packet in the direction of the destination node.

BATMAN employs OGM transmissions as the method of announcing node presence to peers. OGMs are forwarded by neighboring nodes so that distant nodes eventually gain knowledge of other nodes. It is intuitive that this methodology induces flooding in a network; forwarding a frame for every received frame increases bandwidth utilization exponentially. Therefore, BATMAN relies on OGM aggregation, sliding window Sequence Numbers (SN), and variable Time To Live (TTL) values to limit the number of OGMs transmitted.

The TTL field in OLSR varied based on target neighbor distance and purpose; this was done to support MPR election and reduce multi-hop flooding. BATMAN returned to the use of a standard TTL counter that decrements for every forwarded transmission. The default TTL value is 50 hops, though the value is configurable. Since every node in BATMAN may be selected as the best next-hop in a packet flow, all nodes need information about neighbors and distant nodes. The TTL simply restricts the diameter of the network.

Flooding OGMs will eventually overwhelm any medium, given enough nodes. Therefore, BATMAN IV introduced OGM aggregation to improve frame efficiencies. When a node generates an OGM, additional OGMs from neighbors are appended. Upon receiving an OGM from another node, the protocol engine checks the OGM for validity and the newest sequence number for a given origin. New OGMs are queued and then later appended to the local node’s next OGM. At 54 bytes per OGM, a layer 3 packet reaches 98% efficiency by 18 peer OGMs.
To determine the freshness of an OGM, BATMAN uses a Sliding Window algorithm with the sequence number. The sequence number is a 16-bit value incremented for every sourced OGM; when 65535 is reached the SN is rolled over to 0 for the next OGM. BATMAN records the latest SN received from every source. When calculating Transmission Quality, described shortly, packets are evaluated over the previous 64 sequence numbers. Duplicate SNs received from a particular originator are discarded.

Traditional routing protocols rely on the assumption that wired links provide a binary state, connected or disconnected. Therefore, the baud rate of the wired link translates to a metric that is used to evaluate an optimal routing path. In order to support multiple peers on a single interface, full-duplex media require the connection via multiport switch and half-duplex media, such as CDMA links, requires multiple drops on the same line. Regardless of configuration, multi-point wired connections have a level of availability not normally experienced with wireless communications.

Point-to-point wired connections segregate the collision domains (Layer 1 and 2) when multiple nodes form a network, yet the broadcast domain (Layer 3) is effectively contiguous throughout the network due to forwarding at Layer 2. Multi-point wired connections combine the collision and broadcast domains on a single, wired segment. Though bandwidth is restricted on multi-point wired links, to allow for collision detection, all nodes on the segment can detect potential collisions. Scaling the modem’s preamble and the maximum length of the wired link to the propagation latency through the medium ensures collision detection.

Unlike wired connections, wireless links exist on a shared medium that inconsistently combines broadcast and collision domains. The wireless medium complicates the channel access in that links are transient, vary in quality, may be multi-hop in nature, and frequently produce hidden nodes. Therefore, the binary wired link status and bandwidth-based metrics do not correlate well to wireless links.

BATMAN attempts to rectify the inequities of the wireless medium and binary link state protocols by introducing the Transmit Quality (TQ) metric. BATMAN nodes include a TQ value in the
OGM messages to relay the quality of a bi-directional link to peers. In total, there are three TQ concepts in use: Transmit TQ, Receive TQ, and Echo TQ. All three TQ values are calculated on a per peer basis. The Receive TQ is the count of received packets from a 1-hop peer. Echo TQ is the count of a node’s OGMs received due to rebroadcasting from a 1-hop neighbor node. The Transmit TQ for any given 1-hop neighbor is the ratio of the Echo TQ from the peer to the Receive TQ from the peer.

For a source node, when an OGM is created, the TQ value is set to the maximum (255) byte value. A receiving neighbor will multiply its local Transmit TQ, calculated for the source node, with the TQ in the OGM. When the OGM is forwarded, the TQ is replaced by the resultant, providing its neighbor a quality-based metric in the direction to the source.

2.2 Simulating BATMAN

Many of the research papers evaluating BATMAN rely on empirical testing. Though the data is real data, it is inherently stochastic between tests of differing protocols, and even between iterations of the same protocol. Simulations are typically deterministic and allow for little to no variance in test runs. Thus, a more advanced simulator is required; one that can provide deterministic results per scenario yet inject a level of randomness between test iterations to detect the nuances of a protocol.

To validate the findings in [17] [18] [19] [20], the BATMAN IV algorithm was ported to the NS-3 simulation environment [21]. Much of the contemporary research on BATMAN applies to the BATMAN-adv package, having a richer feature set and a more mature code base. Support for the original BATMAN Layer-3 package discontinued in 2009 with the advent of the BATMAN-adv package.

The original BATMAN-0.3.5 package is implemented as a traditional Layer 3 based protocol, where the routing module exchanges UDP packets with peer nodes. In 2007, the BATMAN team began experimenting with a Layer 2 based approach, exchanging frames with peer routing engines and abstracting the physical interfaces from the stack using a virtual BATMAN interface. This paradigm shifted routing decisions from the Kernel’s stack to the BATMAN-adv engine.

Nelson Henry Powell III
The chosen simulation environment is the well-known, open-source NS-3 simulator. The NS-3 simulator is a full stack, event driven simulator supporting various MAC-Phy implementations, networking protocols, transport protocols, and even some applications. NS-3 comes with an OLSRv1 routing package running on top of Layer 3. However, NS-3 does not extend capabilities for virtual interfaces, making a port of BATMAN-adv too lengthy of a process, with too little return value. Therefore, it was reasonable to port the BATMAN-0.3.5 code base as a routing package within NS-3.

![NS-3 Software Architecture](image)

**Figure 2-1: NS-3 Software Architecture**

The NS-3 simulator instantiates all OSI layers 1 through 7 on top of a core simulator package. The Core package provides the smart pointer object model, simulation event core, C++ templates, and supporting macros. Packages at all levels in Figure 2-1 may access the core methods to create events or trace data.

The Network layer comes equipped with various MAC-Phy implementations, such as IEEE 802.11, IEEE 802.15.4, IEEE 802.16, and LTE. Of particular interest for this portion of the research, the 802.11 provides a complete MAC-Phy implementation to include the Ad-hoc mode Mesh capabilities of WiFi. The Internet layer provides both IPv4 and IPv6 stacks and Address Resolution Protocol (ARP) required for the 802.11 ad-hoc mode.
The OLSRv1 implementation exists at the Protocol layer. To implement the BATMAN protocol, the OLSR module was copied and renamed. The core Finite State Machine (FSM) within the OLSR engine is replaced with the FSM straight from the BATMAN 0.3.5 source repository. The routing table update mechanisms, socket operations, and interface configuration remains identical to the OLSR to avoid complications due to rewriting existing software.

Support files for the BATMAN protocol, such as the HNA and data structures files are also copied directly from the BATMAN repository. All code is converted to C++ and when possible, C data structures are replaced with C++ objects to allow for the Object-Oriented Design of the simulator. This method was chosen over implementing sets of void pointer C callback functions. Implementation and registration of C callback functions would require an extensive C to C++ interface-abstraction layer.

For testing the routing protocol, a UDP Client-Server pair would be implemented on certain nodes in the network. The application applied to specific nodes in a test script guaranteed identical network loading between OLSR and BATMAN simulation runs. However, the standard UDP Client-Server applications are lacking in their tracing capabilities.

A new class, called the Aggregator, is introduced to the simulator to support evaluation of the BATMAN protocol. The Aggregator collects information at any level of operation based on registered callbacks from nodes or objects within a node. The Aggregator collects data based into three categories through callback function; categories are named User Data, Routing Data, and Other. As packets or frames are collected for each category, they are included in an additional category called Total, used to determine the aggregate traffic for a given connection. Data may also be written to a file for post processing at the completion of a simulation script.

The type of traffic collected by the Aggregator is specific to the point at which the Aggregator is connected to the simulation. Therefore, an AggregatorHelper class is provided to simplify the connection of an NS-3 container of nodes to a single aggregator. The Helper implements an Install method that checks for various levels of the ns3::NetDevice for protocols, applications, and MAC-Phy interfaces. For each level, the callbacks are applied through the Helper.

Nelson Henry Powell III
The Aggregator can track Packets generated by a UDP Client or Server, packets generated by a Routing Protocol, and Frames transmitted and received at the MAC-Phy layer. When a packet is created at any level of the NS-3 simulator, the packet is always identifiable through the metadata that is attached to the packet. Note that packet metadata is simulator specific and is not counted against the packet in the simulation throughput calculations. When a packet is pushed down or up an OSI layer, the packet is still identifiable by the packet ID which helps associate MAC level frame transmissions and receptions with packet reads and writes.

Since the testing is limited to routing protocol and UDP Server packets, additional frames can only be attributed to 802.11 control frames, ARP Requests, and ARP-Reply frames. Therefore, the Aggregator can determine the amount of overhead required at Layer 2 to provide services to the IP stack. This is critical in determining when and how often handovers occur, causes of routing protocol loops and failures, and pauses in data exchanges.

![Figure 2-2: NS-3 Component Diagram](image)

BATMAN, as written in C, adheres to traditional Linux style lists and data structures. The complex data types were converted to C++ variants when possible. Hash tables were converted to linked lists using the standard library list class. It should be noted that the conversion of the hash tables to a C++ standard list may have introduced priority inversions or incorrect selections when referencing the data.

Nelson Henry Powell III
3 Test Cases

Figure 2-2 provides an overview of the NS-3 architecture when simulating the OLSR routing protocol over an IEEE 802.11 MAC-Phy operating in ad hoc mode. In a similar fashion, the BATMAN code was ported to an NS-3 module. A top-level simulation script was written to automate the iterative test and instantiation of necessary protocols and statistical collection databases.

The first test case was used to determine if the BATMAN routing protocol worked, and if so, what the relative rate of convergence was with respect to OLSR. In the scenario, three nodes are spaced approximately 84m apart linearly. Station pairs \{0,1\} and \{1,2\} have bidirectional communications, but pair \{0,2\} are out of range at 150m. The layout is demonstrated in the NetAnim screen capture in Figure 3-1.

The network is powered up and 2 seconds are allowed to pass to let the protocols kick off their initial messaging. Station 0 is running a UDP Echo Server and Station 2 is running a UDP Echo Client. The UDP Echo Client begins transmitting a 1024-byte packet to the server every 250ms after the first 3 seconds.

![Figure 3-1: Three Node Basic Scenario](image)

Nelson Henry Powell III
After the first 2 seconds, both stations 1 and 2 begin a slow movement, approximately human walking speed, towards station 0. By 12 seconds, stations 1 and 2 reach their closest point and all stations have full bi-directional communications capabilities. The positions are demonstrated in Figure 3-2. At 20 seconds, nodes 1 and 2 begin to move back to their original positions, at a faster movement rate, and complete their movement by 24 seconds.

![Figure 3-2: Three Node Basic Scenario Mid-way Points](image)

Points of interest throughout the simulation include the time till first transmission, time till convergence when \(\{0,2\}\) are in range, and time till re-convergence as \(\{0,2\}\) separate. More subtle items to look for are asymmetric routing paths, route flapping, and routing loops. The UDP Client is configured to transmit a 1024B packet every 250ms. The UDP Server responds to any received packet by retransmitting the original packet back to the client. This results in a maximum of 8KBps offered payload to the system.

The scenario was executed 50 times for both the BATMAN and OLSR protocols. The NS-3 script uses two for-loops to implement Iterations and Runs. A Run through the scenario is a single execution of the scenario with either BATMAN or OLSR as the routing protocol and a specific
random number generator seed. An Iteration consists of two Runs of the scenario, each run using an identical PRNG seed, but different routing protocols.

Figure 3-3 presents the results of 50 iterations of the basic three-node scenario. The Aggregator module groups packets or frames that were transmitted and received into one second bins, and separates the data based on user data, routing data, and total data for every second of the simulation. The data for all runs of a protocol are averaged for each 1-second interval and then plotted using the Python matplotlib library.

The first point of interest is the delay in start of data exchanges between the BATMAN and OLSR protocols. BATMAN converges 1.25sec faster in this scenario, thus BATMAN begins exchanging data by 3sec and OLSR typically does not begin exchanging data until 4.25sec.

A more significant event is revealed as time progresses in the scenario. At approximately 10 seconds, OLSR suffers a complete loss of user data communications. The loss of user data exchanges continues until approximately 27 seconds, where the nodes have nearly reached their original positions. Unlike OLSR, BATMAN maintains the UDP traffic throughout the movement.
except for a short 4 second period. During this period, the BATMAN user throughout is reduced, but not completely lost.

Looking at the raw data, the period where BATMAN drops packets, packets from the Client reach the Server, but Server packets do not reach the Client. This is an indication of an asymmetric routing path. The cause of the asymmetry is due to the slow decay in the Server’s TQ values with respect to the Client. The TQ value directly to the client is so much larger than that of the forwarder node, that it takes approximately 3 seconds for the server to identify that it has lost connectivity with the client.

Though the server takes 3 seconds to identify the loss of communications with the client, the client’s TQ recovers rapidly, and traffic from the client is able switch back to using node 1 as a forwarder. This implies that the TQ algorithm may not be as stable as purported and may be inclined to creating asymmetric routes.

Figure 3-4 provides the four plots as individual graphs and includes standard deviation error bars to demonstrate the skew in performance across the 50 iterations. OLSR’s user data throughput has very little deviation per iteration, which nearly guarantees long service outages during periods of transition. However, OLSR has a periodic, low routing message overhead. When the transition occurs, there is a temporary spike in routing traffic, followed by a 17% increase in steady state routing overhead.

Unlike OLSR, BATMAN is much more volatile. About half way through the first movement phase, there is a short period of volatility. BATMAN has an internal jitter mechanism that randomly offsets OGM generation in an attempt to avoid OGM collisions. As the radios move closer, there are instances wherein collisions occur with OGM or UDP data packets. Later, at 21 seconds, BATMAN may lose user data services or simply suffer from an invalid asymmetric route.

In Figure 3-4, the error bars in the BATMAN User Data graph demonstrates the effects of OGM packet jitter through stochastic testing. The movement pattern does not change across test iterations, however the PRNG stream is unique per iteration. By varying the PRNG, each iteration
produces different transmit times for OGMs within the standard 2 second OGM period. By varying transmit times, reception varies and thus OGM aggregation may be delayed which varies the efficiency of the protocol at the MAC layer.

Inefficiencies at the MAC layer may result in various effects on the system. At Layer 2, inefficient use of the medium results in increased collisions, lost frames, or an increased number of frames sent per available transmit OGM data. The Layer 2 issues thus affect the Layer 3 protocol in that Transmit TQ and Echo TQ values are artificially reduced or skewed, peer OGMs arrive late and cannot be aggregated, and user data starves due to increased routing bandwidth requirements. These effects are demonstrated through the relatively large standard deviation throughout the movement phases (5 seconds through 35 seconds) of the scenario.

The standard deviation is also apparent in the BATMAN Routing bandwidth graph of Figure 3-4. The average throughput is expected to hold around 182Bps. Each node produces a 54-byte OGM with a 20-byte IPv4 header. When aggregated, the maximum packet is expected to be 182-bytes. The error experienced is typically in the negative direction, in that the delayed reception of an OGM from a peer results in a station’s retransmitting the OGM in a separate packet. Since the Aggregator is separating transmitted routing data into 1-second bins, a late OGM could get echoed late enough to have its data accounted for in the next time slice rather than the time slice in which it was originally transmitted.
Figure 3-4: Three Node Basic Scenario Throughput Quad-Plots
Based on the results of Figure 3-4, BATMAN certainly seems promising. Though OLSRs routing bandwidth is periodic and lower than BATMANs, BATMAN does seem to limit the routing traffic required and results in a faster convergence, and thus improved user data performance. But this theory must be tested in other environments to determine the reliability of the results of the first scenario.

Next, the same scenario was retested but with a higher offered user data load. If timing instability affected BATMAN negatively, increasing user data traffic may have additional effects on one or both protocols. For this test, the UDP Echo Client produces a 1500 byte packet every 2ms. With the Echo Server response included, the offered load to the network is approximately 15KBps. The resulting average user data throughput is presented in Figure 3-5.

The initial layout of the curves in Figure 3-5 match the curves of Figure 3-3; however, the recovery of the user data after 21 seconds is dampened for both protocols. BATMAN eventually recovers over a much larger period than before, but it does achieve the sourced 15KBps data rate. OLSR, curiously, never achieves the full 15KBps post convergence.

![Figure 3-5: Three Node-High User Traffic Results](image-url)
The individual curves for data throughput and routing throughput per protocol are presented in Figure 3-5. Unlike the previous test, the OLSR User Data graph shows a high level of error after the 25 second mark. The evaluation of the instability is outside of the scope for this thesis. It is interesting that the OLSR Routing throughput curve looks identical to the OLSR User Data curve in Figure 3-4.

The BATMAN User Data curve has an increase in standard deviation (error) at around 7 seconds. The BATMAN Routing curve looks similar to the previous scenario. Thus, it is assumed that the error in the User Data curve is due to collisions and timed-out packet.
Figure 3-6: Three Node-High User Traffic Quad-Graph
A second more complex scenario was implemented, as displayed in Figure 3-7, to test multiple convergence events for a minimum of 1-hop forwarding of user traffic. In the scenario, the CLI nodes executes the UDP Echo Client, and the SRV node executes the UDP Echo Server. The Client moves around the periphery of the network, never achieving direct connectivity with the Server. The movement is circular and theoretically should result in the Client and Server switching next-hop routers from 10.1.1.3 to the 10.1.1.8, in the order of increasing IPv4 addresses as the Client traverses the physical space.

The combined results of the scenario are presented in Figure 3-8. The first item to note about BATMAN is the notional ceiling on the routing protocol overhead. There are 8 nodes in the system, each generating a 54-byte OGM. With forwarding added to the system, one might expect each node aggregating other OGMs for a combined total of 452-byte packets (including the 20-byte IPv4 overhead). For 8 nodes transmitting 452B at approximately every 2 seconds, the offered routing load should be around 1.8KBps. Instead, the average ceiling holds around 1.2KBps. It is
assumed that the aggregation and other features of BATMAN are at play, capping the maximum routing overhead as described in [11].

An oddity in the graph is the general decline in user throughput for BATMAN while OLSR presents a responsive recovery for each transition, much the opposite of the behavior in the first scenario. Post processing autogenerated Wireshark Pcap files and reviewing NetAnim video produced by the simulator show instances where the next-hop address selection is failing in one or both directions of the traffic flow. The TQ values are not transitioning fast enough to allow for a full hand-over. An excellent example is the movement of the client towards the 10.1.1.4 node. It is apparent that the Server continues to forward traffic through the 10.1.1.3 node while the Client sends traffic through the 10.1.1.4 node. This asymmetry increases as the Client moves across the physical network space.

Figure 3-8: Multi-Node Mobility Results

Figure 3-9 separates the curves in Figure 3-8 for improved readability. The degradation in user data bandwidth shows a high level of variability across the test iterations. Some test iterations, though still degrading performance, do not drop as dramatically as the error bars demonstrate. Regardless, BATMAN is still consistently worse for user data throughput in this scenario.
The data presented does not match the findings in [17] [18] [19] [20]. There are two major sources of error that may cause the inconsistency: porting error, and outdated source code. The most obvious source of error is the conversion from C to C++. Actions such as converting C Hash tables to C++ standard library classes may have inadvertently modified behavior and/or performance. There is also the possibility that the BATMAN 0.3.5 development branch, which is no longer supported, may have inherent bugs. Since the scope of the thesis does not include the debugging and optimization of BATMAN, the cause of the error is not explored.
Figure 3-9: Multi-Node Mobility Quad Graphs
4 Using NS-3 to Simulate Sensors

The NS-3 simulation application was used to compare LEACH and BATSEN in a more realistic environment. NS-3 is the third generation of the Network Simulator. NS-3 is superior to its NS-2 predecessor in that NS-3 provides a higher fidelity, real time, event driven simulation at all levels of the Open Systems Interconnection (OSI) model.

Ns-2 couples C with the Object-TCL/TK scripting language to implement modules. Ns-2 modules are generic abstractions of various layers of the OSI model and do not reflect reality in most cases. NS-3 uses C++ coupled with Python to support an Object Oriented, Smart-Pointer paradigm as the foundation of the simulation package. Frames, packets, and Protocol Data Units (PDU) can be passed to multiple receivers and tracked via callback methods. NS-3 provides a high-fidelity model of each layer of the OSI based on physical reality, kernel system calls, and software APIs.

NS-3 does not provide an “out of box” sensor module, but it does support a well-documented and advanced event driven simulation environment. To test sensors, NS-3 requires the development of a sensor module, a Data Source, and a realistic PHY that can support both CSMA/CA and TDMA channel access methods. For the purposes of this experiment, high fidelity in data sources is not necessary, as frame or packet tracking is all that is required for protocol statistics at this point. The target protocols will function at the sensor module level. Therefore, the first step in simulation is determining the best method to simulate a realistic PHY.

4.1 Physical RF Model Reuse

An NS-3 focus group, mostly individuals from Boeing Corp, implemented a model of the IEEE 802.15.4 2006 standard as the Low Rate WPAN module or lr-wpan. The lr-wpan module implements the four services and Service Access Points (SAP): MAC Data Services (MCPS), MAC Management Services (MLME), PHY Data Services (PD), and PHY Management Services (PLME). The group also created an error rate model based on the 2.4 GHz Additive White Gaussian Noise as prescribed by the IEEE 802.15.4 2006 standard.
As illustrated in Figure 4-1 The lr-wpan-phy class inherits from the NS-3 SpectrumPhy class. Lr-wpan provides for various data rates ranging from 20Kbps to 250Kbps. The 127-byte frames specified in the IEEE standard are identical to the expected frame size in wireless sensors. It should be noted that the PHY does not account for preamble or sacrificial waveform times traditionally inherent in a wideband waveform. The PHY simulates the frame exchange as if they were packets, where the time of reception is the first bit of received payload data or frame header.

The lr-wpan-phy uses a 16-bit MAC address, capable 65,536 addresses. This is considered sufficient for simulation purposes. IEEE 802.15.4 is designed to be a Personal Area Network (PAN), and thus the transmission radius is kept below 100m. It would be impractical to deploy over 65536 nodes in a 100m radius. The address is set through the SetAddr method, which will allow a Sensor MAC to readily deploy unique and trackable PHYs in the simulated environment.

The lr-wpan-phy provides the StartRx, EndRx, PdDataRequest, and EndTx as the primary interface between the channel and the MAC layer objects. All other methods provide various services based on the IEEE 802.15.4 specification. An enumerated type, known as LrWpanPhyEnumeration provides a naming convention for various states of the physical interface, and are reusable for the sensor project.

![Figure 4-1: NS-3 LR-WPAN PHY Class Structures](image-url)
The lr-wpan-phy class provides methods to attach to various channels such as the SingleModeSpectrumChannel. When transmitting, a lr-wpan-phy object creates a PacketBuffer paired with a SpectrumSignalParameter and passes the pair to the channel via the SingleModelSpectrumChannel::StartTx method. This method determines the propagation delay, path loss, and any gain from the transmitting and receiving antennas before scheduling a start of receive event. This process is performed for all nodes active on the channel regardless of transceiver state.

The NS-3 SpectrumPhy uses the NS-3 SpectrumSignalParameters and SpectrumValue classes to perform data transfers and power calculations for the channel. The SpectrumValue object presents an interface to the SpectrumModel class, and implements the frequency-dependent math for PSD aggregation, Signal to Interference and Noise Ratios (SINR), propagation losses, etc. A static SpectrumModel object provides a common method for a developer to implement a list of frequency bins based on the PHY’s spectrum mask.

The SpectrumSignalParameters class contains the pointer to an active SpectrumValue object, the channel the spectrum is operating on, and the model of the transmitting antenna. A node that transmits onto a channel, passes the SpectrumSignalParameters along with packet information such that potential receivers may aggregate the PSD for the given transmission, determine if the signal is strong enough for reception, and locate the received packet reference from the NS-3 core post reception.

It is important that all nodes on a channel process the reception, even if the node is sleeping or transmitting. It is possible that a node transitions from transmit to receive, or from sleeping to receiving. By aggregating signals based on event times, nodes are guaranteed to have an accurate sense of the RF when the transceiver is set to receive. Pre-existing signals act as interference and adjust the SINR accordingly.

The PHY supports a Clear Channel Assessment (CCA) to support the CSMA/CA channel access as defined in the IEEE 802.15.4 section 7.4.2. An existing lr-wpan-csmaca class provides the utility required to perform channel sensing using NS-3 SpectrumPhy constructs. The MAC object
must first set the transceiver state to receiving, or Idle, and then issue the CCA Request. The CSMA module performs a callback to the PHY with a success or failure status, which is then forwarded to an attached MAC protocol.

The 802.15.4 waveform provides for a Slotted and Non-Slotted CSMA/CA protocol. To support various sensor protocols, the unslotted method is used. For generic sensor processing, a method of forcing a transmission with the use of CSMA needs to be created. It was determined that no changes to the Ir-wpan need be made to support such a mode of transmission; rather, the implemented MAC need only modify its sensor-state transitions to support a Forced-TX mode of operation.

The PHY also supports an Energy Detection (ED) to support the CSMA/CA channel access as defined in the IEEE 802.15.4 section 6.9.7. In some cases, 802.15.4 measures the energy on a channel without regard for modulation. The SpectrumPhy aggregates power from all signal sources. When the ED functionality is enabled, the simulator begins sampling the total power on the channel. At every start or stop of a transmission, the simulator incorporates the new aggregate power level to a running average. When the ED process is complete, the module produces an integer value between 0 and 255.

The ED value can be used individually or in conjunction with the CCA feature to detect channel activity. Since this project will not introduce co-site interference from non-802.15.4 transceivers, the ED feature is not required. The ED feature also adds some complexity to the protocol development and provides no immediate value to this project at this time. Therefore, the ED functionality is disabled for the development of the sensor simulation.

The PHY does not detect or deliver SINR, rather it returns a Packet Error Rate (PER) or Link Quality Indicator (LQI), assuming the user implements the error model. If no error model is added in the user script, a perfect LQI is returned every time, and it is up to the random number stream to perform a loss rate on the frames. The authors set the sensitivity of the PHY to -106.58dBm, or 99% success PER for 20B frames.

Nelson Henry Powell III
To support SINR evaluation and information exchanges a LrWpanSnrTag class was developed based on the LrWpanLqiTag. The LrWpanSnrTag inherits from the ns3::Tag class, and can be aggregated as metadata to a packet. The lr-wpan-phy::StartRx and lr-wpan-phy::CheckInterference methods are modified such that a SINR value is derived by dividing the active signal’s power by the total aggregate power on the channel. Power is calculated in milliwatts, and thus Equation 1 is used to determine the SINR of a receive signal.

\[ SINR = 10 \times \log \left( \frac{PWR_{SIG}}{PWR_{TOTAL}} \right) \]

*Equation 1 - SINR Calculation based on Signal Power (mW)*

To determine the maximum testable area within NS-3, as demonstrated in Figure 4-2, the prepackaged test script lr-wpan-error-distance-plot.cc was adjusted to test for 99% success rates on 100-byte frames. The frame length chosen should provide 100% success for smaller control frames at maximal distances and provide a high rate of success when frames are filled to their maximum 127-bytes.

Figure 4-3 presents the resulting PER curves for three power levels: HIGH (-1dBm), MEDIUM (-10dBm), and LOW (-19dBm). Power levels were selected to reduce range approximately 50% per level drop. The final ranges are fixed at 93m, 44m, and 23m respectively. Therefore, all scripts used in the simulation center on a circular layout with \( r = 46.5m \), which ensures a node on the circumference of the test area will have a 1% PER for a node on the opposite side of the area’s circumference.

![Figure 4-2: Simulation Boundaries with Respect to Transmitter Radius](image-url)
The lr-wpan-phy does not model Direct Sequence Spread Spectrum (DSSS) for channel sharing. The LEACH protocol’s TDMA slots are predicated on the use of DSSS to avoid collisions. In lieu of DSSS support, the lr-wpan-phy does allow for up to 13 disparate channels via instantiation of the SpectrumValues class. Therefore, any use of DSSS will be simulated by changing channels rather than using spread codes.

This change is unfortunate, as this does not provide the fidelity desired. Pickholtz describes the need for balancing receive power levels when using DSSS [22]. If receive power levels are not balanced, there is the potential for power loss. LEACH does not provide mechanisms for balancing transmit power, and thus would suffer from destructive interference. This level of fidelity is preferred, but not possible within the scope of this project.

4.2 Sensor MAC Module
For this thesis, the lr-wpan module was copied and renamed as the sensor module. The relationship to the lr-wpan-phy is maintained to avoid rewriting the PHY class. The sensor module begins with the implementation of the SensorNetDevice. The SensorNetDevice inherits from the NetDevice class and implements only necessary virtual functions. The NetDevice abstracts the interface to a network device card, providing a generic interface to layer 3 implementations. The NetDevice also provides the C++ interfaces for object aggregation and by name referencing.

Nelson Henry Powell III
The abstraction of the SensorNetDevice allows for a sensor-specific method and member variables generic for both sensor protocols implemented. The SensorNetDevice contains the specific pointers to the SensorMac, LrWpanPhy, Channel, and CSMA/CA object. Once references are added to the device, associations between the object are made for rapid referencing and interactions.

The SensorMac is an interface, as it provides virtual functions for implementation by child classes. The SensorMac base class allows the SensorNetDevice to contain and interface any MAC child class in a standard manner. The SensorMac provides both a MAC Control Frame transmit queue and Data transmit queue, and a method to determine if a node is dead.

Virtual functions provide the basic interface for starting a protocol’s FSM, reporting node expiration, transmit data success, CCA detection results, transceiver state transition requests, and changing the MAC’s state. It is the responsibility of a protocol to implement the virtual functions to avoid limiting a protocols functionality; but by implementing said virtual functions, the protocol will interoperate with the lr-wpan-phy and higher layers without issue.
4.3 Sensor Helper Class
Like other modules in NS-3, a SensorHelper class is designed to facilitate rapid scripting of large and complex heterogenous networks. Users may define sensor networks of either LEACH or BATSEN protocols, the number of nodes in the sensor network, and attach mobility models, channels, and error models as with most layer 2 classes in NS-3.

The role of the SensorHelper for a SensorMac is extended from the traditional Helper object. The SensorHelper takes on the role of sourcing traffic as well as aggregating statistical information for post execution analysis. There are no sensor application modules in NS-3. Therefore, the SensorHelper provides a ConfigureDataRate method that defines the frequency in which the SensorHelper will schedule events for data delivery to the SensorMac implementations.

![Figure 4-5: NS-3 SensorHelper Class]

As illustrated in Figure 4-5, the SensorHelper also implements a number of callback functions used to track the network’s round, packet transmissions and receptions, and network and node
status. When the simulation is complete, a script may call the AnalyzeData method, which causes the SensorHelper to process and write all analyzed data to text files.

4.4 Power Dissipation
The SensorMac class provides a method known as PlmeSetTRXStateConfirm. This method is configured as a callback from the lr-wpan-phy for every change to the RF hardware’s state. This method is modified to perform system (individual node) power drainage calculations.

A static method variable is used to store the previous time wherein the method is called. When called again, the time delta between calls is calculated as the current time minus the previous time. Note that NS-3 is an event-based simulator, and thus the time values are always correct and are not skewed by the host system’s real-time clock.

The total node power is stored in a member variable called m_totalSystemPower and is measured in milli-Watts. The time delta is multiplied by the current operating power level, and the result is subtracted from the m_totalSystemPower. When the m_totalSystemPower reaches 0mW, the node is considered dead and will not participate in the network.

This approach to power drainage detection is fairly abstract, but it provides enough resolution to complete the project. A higher fidelity to power drainage calculation may be needed if DSSS is ever implemented.
5 The LEACH Protocol

An early adapter of CH based communications is the Near Term Digital Radio (NTDR) developed for the US Army [23] [24]. The NTDR system is a two-tiered hierarchical clustered network approach to wireless networking. NTDR relies on the use of three frequencies and adaptive transmit power to affect an efficient multi-hop network over tens of kilometers. In the early stages of NTDR development, Point-To-Point (PTP) data rates exceeded 500kbps. Later additions to NTDR supported an efficient Multicast traffic routing solution.

The LEACH protocol was an evolutionary step in energy efficient, multi-hop, data forwarding for low energy wireless sensor networks [7]. LEACH expands upon the technological advancements of the NTDR waveform to affect maximal lifetime of the sensor network. The authors define the features of LEACH as:

- Self-Elected CHs
- Adaptive transmit power
- Random rotation of CHs
- In-network processing (compression) of aggregate data
- Scheduled sleep cycles
- Reduced frame contention / collision avoidance through Time Division
- Reduced diameter of network topology
- Complete distributed control of the network

LEACH operates in a four-phase distributed-state machine: Advertisement, Cluster Set-Up, Schedule Creation, and Data Transmission. This state machine, as seen in Figure 5-1, repeats periodically until all sensor nodes run out of power. Each repetition of the state machine is one round. One cycle of the protocol completes when enough rounds have transpired such that every node had an opportunity to function as a CH.
The first phase of the LEACH protocol is CH Advertisement. Every node uses an exponential equation to determine a threshold for the current round and a Random Number Generator to pick a value (inclusively) between 0 and 1. If a node picks a number under the threshold, then the node becomes a CH. Nodes that randomly pick a number greater than the threshold relegate themselves to member node status.

Once the CHs are established, the non-cluster head nodes must request membership with their chosen CHs. During advertisement, all non-CH nodes record the Receive Signal Strength Indicator (RSSI) to determine which CH is closest. The CH producing the highest value, for a given receiver, is the most desirable CH for that receiver. Therefore, a non-CH node will attempt to join with the strongest CH to minimize its own transmit power requirements.

Non-CH nodes must transmit their join request using CSMA. Since there is only one frequency available to the network, the waveform uses spread spectrum transmissions to allow simultaneous transmissions of various clusters with minimal interference. This requires that each self-elected CH must advertise its spread code during the Advertisement Phase, and that each CH selects an orthogonal spread code.
Once the CHs have received all requests for membership, the CHs must create a Transmission Schedule. The schedule implements a temporary Time Division Multiple Access (TDMA) base Medium Access Channel (MAC). The transmission schedule defines when each member node can transmit its data to the CH. Consequently, this also allows the members to derive their transceiver sleep periods, allowing the member nodes to reduce their power expenditure during the Data Transfer phase.

The final phase of normal operations is the Data Transmission phase. During this phase, member nodes wake up at their appointed time, transmit their data to their CH, and then return to a RF sleeping mode. CHs must keep their receivers active throughout the TDMA period, which accounts for a large portion of their energy drain. Data collected from the members is aggregated, compressed, and ultimately transmitted to the network sink node.

Once the Data Transmission phase completes, the next round in the waveform begins by restarting the Advertisement Phase. All CH nodes from preceding rounds, in the current cycle, are no longer eligible for self-election, thereby reducing the pool of nodes to compete for CH status. After all nodes have had an opportunity to operate as CHs, the protocol starts a new cycle with the 0th round.

In order to achieve power savings, the Setup-Phase must be significantly shorter than the Steady-State Phase. The first three states in the FSM in Figure 5-1 constitute the Setup-Phase. During these states, all nodes must be awake and exchange information in preparation for the data exchanges. The last state in the FSM is considered the Steady-State Phase; this is the state wherein nodes are allowed to sleep to conserve energy.

### 5.1 Benefits of LEACH

Unlike NTDR, LEACH rotates CH responsibilities periodically. Distributing the CH responsibility across all nodes equally minimizes the average power expended, and thus extends the life of the sensor network. According to the results of the original LEACH research paper, the protocol optimizes power utilization when 5% of the nodes become CHs during the Advertisement phase.
of the protocol, and that all nodes get a chance to be a CH once and only once in a cycle. As an example, for a 100-node network, one cycle would consist of 20 rounds with 5 CHs per round.

CH membership attempts to further reduce energy expenditure by co-locating members to CHs. During the Advertisement Phase, the non-CH nodes record the RSSI value received from each prospective CH. It is assumed, by the receiving nodes, that the higher the RSSI the closer the CH node. Thus, all non-CH nodes attempt to join the clusters whose CH RSSI was highest during the Advertisement Phase. Relying on proximity allows for a reduction in non-CH transmit power for additional energy savings.

The Data Transfer Phase reduces energy further by allowing non-CH nodes to disable their transceivers during the TDMA slots of other members. The CHs announce the TDMA schedule at the end of the Cluster Set-Up phase. Members of the cluster do not need to receive data from peer nodes, and thus can safely power down the RF circuitry until it is their time to transmit or the end of the TDMA schedule is reached.

Another optimization for the reduction in power consumption is data aggregation and compression at CHs. Every member node transmits its data to its associated CH. Each CH aggregates data from its members, compresses the data, and then transfers the data to the sink. This reduces the need for high powered direct transmission by all member nodes, in that members attempt to transmit to the closest CH possible.

Finally, LEACH uses Code Division Multiple Access (CDMA) or Spread Spectrum, DSSS specifically, to share the frequency amongst multiple Clusters. During the Data Transfer Phase, each cluster uses an orthogonal spreading code such that only one frequency is required. This allows nodes to share the same TDMA time slot, yet avoid collisions. This also supports secure transmissions amongst cluster members only.

5.2 Converting NS-2 Baseline Code
Heinzelman et. al., implemented an improved version of the original LEACH protocol using the NS-2 simulation environment. The code, developed as part of the MIT uAMPS project, included

Nelson Henry Powell III
its own MAC/PHY model sourced by an OTcl abstraction of the sensor nodes [25]. Other authors of subsequent research on LEACH rely on MATLAB and NS-2 to replicate original results as well as test new theories [26] [27] [28] [29].

LEACH attempts to optimize energy utilization for two-hop forwarding of sensor data. According to the original paper, the LEACH protocol outperforms both Minimum Transmit Energy (MTE) routing protocols and Direct Communications networks. MTE based networks are those wherein nodes attempt to use minimal transmit power to forward data across multiple hops to a sink node. Direct Communications networks are ones in which all nodes in a sensor network attempt to communicate with a sink without the need of intermediary nodes.

To reap the benefits of the original LEACH protocol, the authors make several assumptions regarding sensor node functions and their supporting waveforms. Some assumptions are explicitly stated while others are implied, based on the way the research was performed. The following list summarizes the assumptions identified in the original paper [7] [25]:

1) No mobility in the system
2) Node count known a priori
3) All links are always bidirectional and symmetric
4) Geography of the network is always less than the radius of a transmitter’s capability
5) RSSI accurately represents distance
6) Topology is uniformly distributed
7) Spread spectrum guarantees no interference during the Data Transmission phase
8) All nodes make successful decisions
9) Setup-Phase is relatively small compared to Steady-State phase

5.2.1 Assumption 1
The original LEACH paper does not account for node mobility. Mobility adds a level of complexity in any routing algorithm, as well as test platform requirements, in that the waveform performance becomes both time and space dependent. Moreover, the lack of mobility ensures that the sensor application produces geospatially related data sets. LEACH assumes that similar data

Nelson Henry Powell III
sets easily compress or may be dropped if there is a strong correlation of the data based on physical locations. This reduces the aggregated data payload for CH to Base Station communications [7].

5.2.2 Assumption 2
LEACH results reveal the optimal CH assignment ratio as 5% of the network size (N). In order for the protocol to work in the real world, the waveform must provide a node announcement phase, wherein nodes broadcast their presence allowing all nodes to get an accurate node count. This requires that all nodes are within broadcast range of all other nodes. If there is no announcement phase, the node count must be configured pre-deployment [25].

Waveforms face a tradeoff when implementing dynamic membership. If the waveform prefers dynamic membership, then additional time must be allocated to the network for presence messaging, thereby reducing energy savings. If nodes are unsuccessful in announcing their presence, there may be disagreement within the nodes as to the number of nodes, and thus what constitutes 5% of the network for CH reservation. This usually leads to either centralized membership control at the sink, or the protocol must be modified to compensate and distribute the membership logic.

On the positive side of dynamic membership, mobility and Late Net Entry (LNE) are natural progressions to the protocol’s development. Regardless if the network is centrally controlled or distributed, having LNE mechanisms allows for presence identification early in deployment as well as later when dead nodes are replaced with fresh sensors. A well-architected LNE mechanism can also be modified for secure net entry.

5.2.3 Assumption 3
The authors of LEACH assume that all RF links are bidirectional and symmetric [7] [25]. Bidirectional links imply that any two nodes can both successfully transmit to and receive from its peer. Symmetry refers to the idea that the SINR for any transmission is similar if not the same in both directions for any given pair of nodes. This is successful in the original simulation in that there is no movement and there is no noise floor. The updated uAmps project leverages Multi-
Path Fading and Path-Loss models, but these models are implemented in NS-2 which has very low fidelity compared to the real-world [25].

Symmetry requires that all sensor nodes have identical power and transmit capabilities, are within RF range of all other nodes, and that the radius of the network is never larger than the maximum transmit range. This assumption also requires static placement of nodes to avoid movement beyond the M x M region limits. Lack of symmetry may result in incorrect CH selection, failure to join a cluster, or loss of control messaging throughout a round.

5.2.4 Assumption 4
The original LEACH research paper used a square for the testable RF region [7] [25]. Using a square test region may unexpectedly affect test results based on the node’s maximum transmission range. If the sensor’s RF is capable of transmitting from corner to corner along a square’s bisecting angle, then the sensor network will be compressed. If the RF is only capable of reaching an adjacent corner in the square, there will be dead space outside the circumference of the transmitting node’s RF range, or the network is dilated.

In the case of network compression, the radius of the maximum power level transmission (r) is equivalent to the length of the hypotenuse of the bisecting angle (c), where \( a^2 + a^2 = c^2 \). Figure 5-2 demonstrates various locations of a node within the test square. The three concentric circles indicate the power levels as they reduce from High, Medium, and Low. Relative to the RF range, nodes will be physically clustered making it difficult to distinguish range or variations in RSSI.

![Diagram showing network compression when R = C](image)

Figure 5-2: Network Compression when R = C
When nodes are located in the center of the test region, Figure 5-2(C), there is an extremely low probability they will ever need to use power greater than a Low-Power setting, as only nodes in the corners need Medium-Power. Therefore, the closer to the center a node is positioned, the less power it uses throughout the lifetime of the network.

Nodes positioned in the far corners of the test area, Figure 5-2(A), are the only nodes that will ever use a High-Power setting. This is caused by the need for every node to reach all other nodes in the network. If the Sink is placed in the corner of the test space, then nodes in the opposite corner have an unfair disadvantage with respect to power conservation.

In the case of network dilation, the radius of the maximum power level ($r$) is equivalent to the length of the square’s side ($a$). As illustrated in Figure 5-3(C), nodes closer to the center of the network can operate at most two distinct power levels in their PHY. However, nodes existing at opposite corners of the square, Figure 5-3(A), will never hear from one another. This causes a hidden node problem which is outside the scope of the original LEACH protocol specification, as well as this thesis.

As presented in Figure 4-2, for a node to reach the farthest point in a simulation, the radius of the node’s transmission must be at least as long as the bisecting line between two corners of the square. Stated mathematically, the simulations maximum boundary length must be no more than \( \sin(45) \times R \) or \( 0.7071R \). But to maximize the testable space without artificially compressing the network, a circular test area should be used.
As illustrated in Figure 5-4, by setting the test area’s diameter \( (d) \) to the transmitter’s maximum range \( (R) \), nodes toward the outer edge are always able to utilize the highest power if needed and there is no spatially induced hidden node. This also utilizes space outside the square normally not accessed in typical testbeds. This layout still reduces nodes in the center to a dual-power mode, but the probability of using three power levels increases rapidly upon node placement outside of the center.

5.2.5 Assumption 5
In keeping with the notion of static placement within a fixed radius, the Receive Signal Strength Indication (RSSI) values of all receptions are proportional to the distance between the transmitter and receiver [7] [25] [26] [27]. In real RF communications, this is not always true, but the assumption allows individual nodes to determine the best CH candidate based on proximity [22] [30]. Tied in with the variable transmit power capabilities, this assumption helps reduce power consumption in that nodes always register with the closest CH.

This assumption gets complicated when considering the seventh assumption, utilizing DSSS to provide channel reuse. As noted in [22], if transmit power between any two transmitting nodes is not balanced by a receiver, then the simultaneous reception may result in destructive or constructive interference, thus artificially shifting the RSSI value. Use of an RSSI value, without considering the Signal to Interference and Noise Ratio (SINR), may result in invalid distance estimates, and quite possibly, selection of the wrong CH for a given node.
5.2.6 Assumption 6
Not only are the nodes confined within an M x M region, node distribution must be relatively uniform [25]. As nodes always attempt to join the closest CH, unequal distribution of nodes would result in uneven distribution of power drain. Using a uniform, random distribution of nodes ensures an optimal power distribution across the network. This may not be practical in a real-world deployment, but such distribution assists in determining a theoretical maximum.

Clusters should contain the same number of members. If the sizes of clusters vary, then some CHs will expend more energy than others. The fundamental purpose behind LEACH is that energy expenditure is distributed across the network evenly; thus, the waveform should either force the number of nodes per cluster to be the same every round or the nodes should be distributed equally to naturally associate in near equally sized clusters.

Cluster membership must also be uniformly distributed based on distance. If membership is not restricted in distance, the advantage to in-network processing of correlated data sets is diminished. Furthermore, the power required to reach the CH might exceed the power required to reach the Sink directly, and thus cause unnecessary power drain [27] [26].

Since LEACH uses a per node random number to determine the CH selection, there is no way to guarantee the distribution of CHs geographically. It is assumed that statistically, the distribution will average out over time. With the original LEACH code, the random source is centralized in the simulator, and thus, even distribution may be artificially created [7] [25]. Since the NS-2 variant of the LEACH code was not tested, this source of error is not explored.

Another issue with this assumption is that there is no guarantee that the protocol results in an even distribution of CHs throughout the test space [26] [28]. If CHs elected in the same round are co-located, it would be difficult to perform the in-network processing of data sets. Cluster members are more likely to be physically distant from the CHs, and thus their data will not correlate to the CH’s data sets.
5.2.7 Assumption 7
During the Data Transmission phase of operation, each cluster uses an orthogonal spreading code to ensure spectrum reuse. Spread codes are either prepopulated per node, or nodes must select a code randomly from a pre-populated list [25]. Assigning individual spread-codes may not be a scalable approach to DSSS, so the natural tendency is to randomly select a spread-code from a pre-programmed list of codes.

Selecting a code randomly does not guarantee unique spread-code selection between any two CHs. If two CHs select the same spread-code, then all TDMA slots will experience destructive interference. If cluster members are all physically close to their CH, and CHs with the same spread-code are far apart, then it may be possible that a reduction in member transmit power will allow for reception of TDMA transmissions. However, based on assumptions 4 and 6, this is most likely not the case.

Another alternative to self-selection is centrally controlled, or Sink allocated, spread-codes. In this case, the Sink assigns unique spread codes to ensure the orthogonality of all clusters. This fixes the previous problem, but eliminates the decentralized control, which is fundamental to the LEACH protocol [7].

The IEEE 802.15.4 2003 standard specifies two physical layers supporting DSSS. For nodes supporting the IEEE 802.15.4 PHY, LEACH may be a plausible MAC. However, there is a price to pay for DSSS. When there are multiple transmissions using DSSS, there is typically a loss in power, which reduces the range of the transceiver [22] [30]. In order for DSSS to support the original authors’ theory, transmitters must be synchronized [25], and transmit powers must be balanced for each receiver [25] [30]. DSSS also requires more power based on the wideband spread, which must be accounted for in both simulation and the real-world [25].

5.2.8 Assumption 8
The original LEACH algorithm’s implementation ensures all nodes make the correct decision and get what the network wants [7] [25] [26] [27]. In all simulations, individual nodes are omniscient to various faults.
The LEACH paper expresses the CH selection algorithm based on a random value. However, in a real-world distributed system, no two nodes will resolve the same random value for a given round. There is a probability that too many or too few nodes will attempt to become CHs in the same round. One might use pre-planned CH assignments per cycle, centralized control from the Sink, or an election detection process to avoid this error.

Using the prescribed 5% of the network as CHs, and the original formula for self-election, the probability of node self-election is plotted in Figure 5-5 for a network with 100 nodes. From the graph, nodes are highly unlikely to self-elect for most of the rounds in a given cycle of the protocol. Without an omniscient simulator distributing the PRNG to the nodes, the protocol will produce multiple rounds with fewer than required or even no CHs.

![Figure 5-5: Probability of Node's Self-Election per Round](image)

Another implementation detail is that non-CH nodes are always assigned to CHs evenly. This ensures that all nodes are attempting to use the minimum transmit energy to reach their CH. This also guarantees that the TDMA cycles are kept to a minimum: the number of non-CH nodes divided by the number of CHs. Though this provides a reduction in power in comparison to conventional sensor routing algorithms, the solution may still cause excessive power loss. If nodes
are closer to the Sink than they are to their assigned CH, it may be able to save energy by sending data directly to the Sink at a reduced power level.

Another item in the original LEACH papers is that there is no acknowledgement to a Join-REQ sent by a node to a target CH [7] [25]. During the Cluster Setup phase, nodes transmit a Join-Request to their preferred CH using CSMA. Even with CSMA/CA, there is no guarantee that CHs will receive all Join-REQ frames. Failure to acknowledge this behavior implies a level of simulation omniscience that does not translate to a real-world network.

5.2.9 Assumption 9
Finally, all implementations of LEACH assume that the Setup-Phase is much smaller than Steady-State [7] [25] [26] [27]. The Setup-Phase is considered the first two states in the FSM in Figure 5-1, while the Steady-State is composed of the last two states in the FSM. Nodes only perform the RF sleep functions during the Steady-State operations, and thus the power savings derives from the disproportionate time spent sleeping.

It was noted that simulations of LEACH do not account for many of the aforementioned assumptions, and that the Setup-Phase is assumed in simulation [25]. Ignoring network management requirements diminishes the fidelity and relevance of a simulated protocol. To realistically implement LEACH and compare it to other protocol types, one must account for all aspects of the protocol, not just the optimizations.

5.3 A LEACH Protocol Specification
The LEACH implementation began with the simple FSM as illustrated by Figure 5-1. To align the LEACH protocol in NS-3 with the original specification, the new LEACH FSM overlays the original states as demonstrated in Figure 5-6. There were multiple attempts at defining the LEACH FSM using a near-real PHY; the FSM in Figure 5-6 is the culmination of multiple attempts to implement the FSM based on the original NS-2 code and the reaction to failures cause by the tacit assumptions of the original design.
Based on the LEACH protocol defined in [7], the original NS-2 code [25], and realistic waveform development practices, the LEACH protocol is realized using NS-3 and the existing lr-wpan module. The following sections describe the Finite State Machine implemented to support a functional LEACH waveform and the timing analysis of the implemented protocol to support post-data collection analysis.

### 5.3.1 State Machine Description

The original FSM in Figure 5-1, combined with the adjustments needed to account for the assumptions in section 5.2, result in the implemented FSM illustrated in Figure 5-6. The phases of the new FSM are overlaid on top of the original FSM to demonstrate adherence to the original design. The FSMs only illustrate the member node behavior. Sink behavior is described concurrently with the non-Sink node states.

![Figure 5-6: LEACH Finite State Machine](image-url)
5.3.1.1 Initialization State

To begin the design, a fifth state, the Initialization State, had to be added. The LEACH design does not prescribe the use of timers or signaling to control state transitions. In NS-2, all nodes simply transition to the next round at the same time. To maintain tight synchronization, the Initialization State provides a time for all nodes to passively listen for a LEACH_INIT frame from the Sink, indicating the start of the next round. Once received, all nodes move into the Advertisement Phase as shown in transition 1 in Figure 5-6. Referring to [25], time synchronization aids in the use of DSSS and TDMA slot alignment.

5.3.1.2 Advertisement State

Once in the Advertisement phase, every node rolls a random number to determine if it should act as a Cluster-Head. After implementing LEACH for the first time, it was discovered that offering a single attempt at CH Advertisement was impractical. Using a uniformly distributed random number generator with a unique seed for each node resulted in too few nodes electing themselves as CHs early in a cycle. This forces a large number of nodes to transmit their data directly to the Sink, or operate as Direct Connections (DC), in the early rounds of the protocol’s lifecycle. In later rounds, when the random number threshold increases exponentially, there is a significant spike in CH assignments. It was assumed, in accordance with [7], that heavy reliance on DC transmissions would result in too much power loss.

Therefore, an additional message type (ADD_MORE_CHS) was created to support a second attempt at CH Advertisement. In the first attempt at advertisements, all self-elected CHs transmit the CH_ADV frame. The Sink records the nodes as potential CHs. Collisions may occur, as the CH_ADV messages are transmitted using CSMA/CA. Thus, there is a probability that the Sink may miss multiple CH_ADV messages.

When the Advertisement Phase timer in the Sink times out, the Sink will check to see if the required node count for CHs is met. Note that only the Sink has an Advertisement Phase timeout, as all nodes are subject to the Sink’s direction. If there are not enough potential CHs, the Sink will transmit an ADD_MORE_CHS frame. The reception of the ADD_MORE_CHS is demonstrated as transition 2 in Figure 5-6.
In the original LEACH protocol, nodes use the algorithm Equation 2 to determine if they should become a CH for the next round. In the algorithm, $P$ is the desired percentage of nodes that should be CHs and $r$ is the current round number. The $\text{Mod}(r, \frac{1}{P})$ portion of the equation ensures that the process is repeated every $\frac{N}{P}$ rounds. The random number is selected from a range of $R[0,1]$, and if less than $T$, the node elects itself as a CH. As $r$ approaches $\frac{N}{P}$ rounds, the probability of a node self-electing to CH status increases exponentially. Once a node becomes a CH, it cannot become a CH again until the $\frac{N}{P}$ rounds have ended and a new cycle begins.

$$T = \begin{cases} 
\frac{P}{1 - P \times \text{Mod}(r, \frac{1}{P})} \\
0 
\end{cases}$$

*Equation 2 - LEACH Cluster Head Selection Threshold*

One caveat to the original threshold equation is that, realistically, each node has a different PRNG seed. Therefore, it is highly probable that the desired CH population is not realizable for early rounds in each cycle. Therefore, the ADD_MORE_CHS frame primitive was added to force nodes to re-evaluate their CH status. Nodes that have already self-elected as CHs do nothing, while non-CH nodes would run the algorithm a second time to increase the number of self-elected CHs.

Using the same algorithm proved futile, as the PRNGs typically failed a second time to acquire enough potential CHs to reach the $P$ percentage. Therefore, the algorithm was modified such that the same threshold $T$ is calculated, but that the PRNG used a range of $[0…M]$ where $M$ is defined as $M = T \times (\frac{1}{P} \times 0.9)$ instead of the traditional $[0…1]$. By adjusting the top end of the PRNG to $M$, as a multiple of $T$, the threshold, though exponentially increasing per round, is held at a constant ratio of $M$. This improves the probability of nodes in early rounds for self-electing as CHs but does not improve the chances of nodes in later rounds.

The second caveat regarding the original threshold calculation is the fact that unless the number of desired cluster heads is a common denominator of the number of nodes, the number of rounds per cycle is not an integer nor is a threshold of 1.0 achievable. It is unknown how the original...
authors adjusted for this. In this simulation, the first call for CHs remained true to the original authors’ equation. The number of rounds per cycle is calculated by truncating the result of dividing the number of nodes by the number of desired cluster heads. This causes a shortage of rounds per cycle, but conserves energy in that some nodes will not act as CHs during a given cycle.

5.3.1.3 **Cluster Setup Phase**

If the Sink receives enough CH_AdVs in the first advertisement attempt or completes a second advertisement timeout regardless of CH_AdVs received, the Sink transmits a Final CH List (RX_FINAL_CH_LIST) message. The frame contains the Sink’s list of selected CHs and their assign channel. Note that the original LEACH specification uses DSSS to separate CH cluster transmissions. The lr-wpan module does not implement a DSSS PHY, and thus to simulate DSSS, each CH and member nodes change channels to avoid interference.

To avoid additional transmissions or pre-deployment configuration variables, the Sink provides a unique channel (or DSSS spread code in the case of the original specification) associated with each of the CHs in the RX_FINAL_CH_LIST message. When transitioning to the Cluster Setup state, the CHs change channels and listen for Join Request (JOIN_REQ) messages from potential member nodes.

Nodes that have not elected themselves as potential CHs follow transition 3 into the Node Schedule state, which implements the original Cluster Setup Phase; similarly, nodes that had elected themselves as potential CHs, but are not in the Sink’s list of CHs follow transition 4 into the Node Schedule state. Nodes that elected themselves as potential CHs and were selected by the Sink, follow transition 5 to the CH Schedule state.

While in the Cluster Setup Phase, selected CHs wait for JOIN_REQ messages from the non-CH nodes for a fixed time-period. The timeout is a multiple of the time it takes to transmit a frame and reply with an acknowledgement (ACK) and the number of nodes, less the number of nodes assigned as cluster heads.
Non-CH nodes, having received the RX_FINAL_CH_LIST message, select the closest CH via best SINR. Each node changes channels from the primary Sink channel to their desired CH channel to transmit a JOIN_REQ message. Non-CH nodes use the same timeout calculated by the CHs to randomly select a transmit time in which to send their JOIN_REQ.

Upon receiving a JOIN_REQ, the CH determines if there are available slots to give one to the requesting node. If a slot exists, a Join Acknowledgement (JOIN_ACK) is sent back to the requesting node. If there are no more slots, a negative acknowledgement (JOIN_NACK) is sent to the requesting node. The CH’s response to the JOIN_REQ is presented as transition 6 in Figure 5-6.

If the node sending the JOIN_REQ receives a JOIN_ACK, the node goes to sleep until the end of the Cluster Setup Phase, transition 7 in Figure 5-6. If the node receives a JOIN_NACK, transition 8 in Figure 5-6, the node assumes that the TDMA schedule is full and must select the next best CH (by SINR ranking) to join. The node changes channels and repeats the JOIN_REQ process in the time remaining.

It is possible that two nodes collide when transmitting their JOIN_REQ messages. One node might receive an ACK for the other node when listening for its own ACK. When this occurs, the node must assume that the CH did not receive its request. Therefore, the node must reschedule its transmission for a second attempt.

5.3.1.4 Schedule Creation Phase
Once the timeout expires, all nodes transition to the Schedule Creation phase as shown in transitions 9 and 10 in Figure 5-6. CH nodes return to the primary Sink channel to wait for the Sink to signal the start of the TDMA data transfers via the TDMA_KICK_OFF message. Reception of the message helps synchronize the start of the TDMA period, as the distance limit as defined by the maximum transmit power dictates the maximum propagation delay. The Sink begins a TDMA timeout timer and adds a 2x time of propagation delay assuming a CH could be the furthest distance from the Sink, and this starts the TDMA period late.

Nelson Henry Powell III
Every CH node returns to its assigned channel and transmits the ADVERTISE_SCHEDULE message. This message contains the list of all accepted member nodes, in the order of TDMA slot assignments. Non-CH nodes that did not get a slot in their selected CH’s TDMA list follow transition 11 to the Direct Connect state; similarly, non-CH nodes that never received a JOIN_ACK also follow transition 11 to the Direct Connect state. Nodes that receive a TDMA slot assignment transition to the Node Data state, while all CHs transition to the CH Data state.

5.3.1.5 Data Transmission Phase
The Data Transmission Phase from the original LEACH specification is broken into two sub-phases in the implemented protocol. First, all non-CH nodes must transmit sensor data to their CH in their assigned TDMA slot. After the TDMA transactions are complete, all CHs must transmit the aggregated data to the Sink. To support Direct Connect nodes, DC nodes must transmit their data to the Sink during the Data Transmission phase.

It should be noted that 2x the maximum propagation delay is added to the minimum time to transmit a frame to account for instances when a non-CH node must reach a CH node across the simulation space. By offsetting transmissions by 1x propagation delay, the protocol ensures that the tail end of a long-range transmission does not collide with a short-range transmission that succeeds the previous. This added time also increases the sleep time for all member nodes marginally.

Since Direct Connect (DC) nodes are aware of the maximum TDMA slot time, every DC node will pick a random time within the TDMA period to attempt a direct transmission of data to the Sink. Since there is no a priori TDMA slot assignment, DC nodes resort to CSMA/CA for a best effort delivery of data.

For the purposes of simulation, the data transmitted from any sensor node is 100 bytes of random data with a globally unique sequence number. The payload is inconsequential for this thesis, but the payload length does affect the results. The sequence number is tracked by the Aggregator module to determine packet success rates and network throughput rates.
When a CH receives the data from each member node, it does not aggregate the data; rather the CH node aggregates a list of member nodes from which it successfully received a data frame. The CHs also keep a list of all global sequence numbers received so that the Aggregator can compare Sink-received sequence numbers with the transmitted sequence numbers.

In the second half of the Data Transmission phase, the CHs transmit a special data frame, wherein the data is composed of MAC addresses from which the CH received data payloads and the globally unique sequence numbers associated with said receptions. Additional successive frames are sent if there are too many MAC addresses and sequence numbers to fit in a single data frame. MAC addresses are 2B and sequence numbers are 4B. Thus, the maximum number of required packets per CH is known by all nodes at network configuration time calculated by Equation 3.

$$F_{COUNT} = \frac{(Frame_{MAX}-(Header+Footer))}{6 \text{ Bytes}}$$

*Equation 3 – Maximum Number of Nodes in a Data Frame Payload*

To some degree, testing data delivery in this method is unrealistic. However, the original LEACH authors’ contention was that co-located nodes would have similar data sets. It has yet to be proven that this is a realistic assumption. More importantly, it is questionable if member nodes are always co-located with their assigned CH. The choice to deliver aggregated MAC addresses and sequence numbers simplified the NS-3 callback interfaces and frame identification in the Aggregator module.

### 5.3.2 Protocol Timing Analysis

The length of a LEACH round can be calculated by laying out the FSM states into a timeline and adding the time periods as defined from the perspective of the Sink. Time-periods are defined as a function of the number of nodes \((N)\) and number of CHs \((C)\). Time constants or constant multipliers are literally defined within each phase’s time-period definition. Slot lengths are defined as either CSMA/CA based or TDMA based, where \(T_{CSMA}\) is 7ms and \(T_{TDMA}\) is 5.2ms. The \(T_{CSMA}\) slot occupies approximately 5.2ms for transmitting a maximum frame (127 Bytes) and another 1.8ms for the carrier sense function. The following timing description expands upon the timeline diagram in Figure 5-7.
The first two advertisement phases use an identical calculation expressed in Equation 4(A). During the Advertisement phases, nodes wishing to become CHs use CSMA/CA to broadcast their intent. The CSMA/CA process does not guarantee access, merely avoids noticeable collisions. The time required to transmit a frame plus the channel activity detection, or $T_{CSMA}$, is about 7ms. Knowing that the protocol requires a fixed number of CHs, the number of CHs is multiplied by $T_{CSMA}$. Using a random number generator, a random start time is selected within the $T_{ADV}$, starting from the reception of the Super Frame Announcement.

$$T_{ADV} = (C \times T_{CSMA}) \quad (A)$$

$$T_{ADV} = (M_1 \times C \times T_{CSMA}) \quad (B)$$

**Equation 4 – Advertisement Phase Timeout Calculations**

Multiplying the CH and $T_{SLOT}$ does not provide enough time to reliably avoid collisions. The probability that any two nodes, transmitting their intent to become cluster heads, have a collision is $\rho = 1/C$. Therefore, a constant multiplier is used to increase the time-period sufficiently to reduce the probability of collisions, as expressed in Equation 4(B). For the purposes of implementation, the constant multiplier $M_1$ is set to 3.

During the Scheduling phase, all non-CH nodes must attempt to join a cluster and use a time-period to transmit their intent presented by Equation 5(A). Again, all transmissions are performed using CSMA/CA. Nodes must transmit a JOIN request and wait for an ACK or NACK frame from the prospective CH. The initial JOIN request takes at least the standard $T_{CSMA}$ time due to the channel sense; the ACK or NACK is transmitted immediately after the JOIN reception, and thus takes less time than $T_{CSMA}$. As a rough approximation, at most 2x $T_{CSMA}$ is needed to complete a single node’s join request process.

$$T_{SCHED} = (N - C) \times T_{CSMA} \quad (A)$$

$$T_{SCHED} = M_2 \times (N - C) \times T_{CSMA} \quad (B)$$

**Equation 5 – Scheduling Phase Timeout Calculations**
It is possible that some nodes are not authorized to join their desired CH or never receive an ACK from their chosen CH. In these cases, a node may need to change channels and attempt to join an alternate cluster. Therefore, additional time is needed to support the contingency join operation. This process, in addition to the Join-Ack timing, forces the time to dilate as expressed in Equation 5(B). The governing constant $M_2$ takes both processes into account. The constant multiplier used in this implementation is 2.7.

The Data Phase is the most efficient phase of the protocol in that the CHs transmit a TDMA schedule allowing nodes to sleep. Moreover, without the need for channel sensing prior to transmission, the time required to transmit is reduced to 5.2ms or $T_{TDMA}$. If the protocol functioned optimally, the time-period of the Data Phase could be defined in Equation 6(A).

Unfortunately, there are three inefficiencies in this state that must be accounted for. First, the protocol must support a single CH operation. In some cases, only one node might elect itself as a CH; this happens more often towards the end of a cycle. If there’s only one CH, but the required CH counter is greater than one, the single CH must be able to handle all nodes in the network.

Similarly, nodes die off over time and may cause a loss of CHs. Though 5% of the nodes is the preferred CH to Node ratio, as a network ages and nodes die, there will come a point where only one node is active as a CH. In this case, a node must be able to acknowledge and support the entire network in the absence of other CHs.

$$T_{DATA} = (N - C) \times T_{TDMA} \quad (A)$$

$$T_{DATA} = (N + C + x) \times T_{TDMA} \quad (B)$$

Equation 6 – Data Phase Timeout Calculations

The third issue occurs when there are no CHs elected and all nodes operate in a Direct Connection mode. This requires that there is enough time for every node in the network to transmit directly to the Sink. Unfortunately, the Direct Connection method relies on CSMA/CA operations and does not guarantee data delivery. Therefore, there must be enough time for all nodes to perform the channel sense and transmit their data frame. Since the shift to full network Direct Connections
occurs most often later in the lifetime of the network, and nodes die in the later portion of the
network lifetime, the $T_{DATA}$ does not require a large increase in time.

Equation 6(B) expresses the final algorithm required to determine the complete TDMA slot
assignments. The final Data Phase timeout multiplies the number of nodes ($N$) with the TDMA
slot length ($T_{TDMA}$). To allow for extra time, the number of CHs is doubly counted by adding $C$
to $N$ prior to the multiplication.

An extra slot ($x$) is added or subtracted in Equation 6(B) based on node identity. If the node is a
Sink, a slot is added since the Sink must transmit the TDMA_KICK_OFF frame before the Data
Phase begins. CHs add no additional time, as the TDMA process begins with their
ADVERTISE_SCHEDULE frame transmission. All Cluster Member nodes and Direct Connect
nodes subtract one $T_{TDMA}$ slot, as they must align their sleep time with the CH’s TDMA timeline.

The Sink Data Phase is the point in the process where all CHs transmit their aggregated data to the
Sink. Each CH may have collected Data Frames in the preceding phase. For the simulated
environment, there is no sensor data, rather the data is just a list of MAC addresses and sequence
numbers for tracking transmission success rates. The Sink Data Frame is composed of a list of
MAC addresses with their globally unique sequence numbers for the given round.

With a maximum payload of 114 bytes, and 6 bytes per peer data set, a CH must transmit one
frame for every 19 data frames received. Therefore, the maximum number of data frames a CH
may transmit is calculated in Equation 7(A). When the network has a single CH active, the CH
must forward for all other nodes; thus, the number of member nodes ($n$) becomes the total number
of nodes in the network, or ($n \rightarrow N$). To avoid additional messaging, the worst case is always
assumed for the Sink Data Phase. The entire Sink Data Phase is expressed in Equation 7(B).

$$F_{MAX} = \left\lceil \frac{(n \times 6)}{PL_{MAX}} \right\rceil$$  \hspace{1cm} (A)

$$T_{SINK} = C \times F_{MAX} \times T_{TDMA}$$  \hspace{1cm} (B)

$$T_{START} = CH_{POS} \times F_{MAX} \times T_{TDMA}$$  \hspace{1cm} (C)

Equation 7 – Sink Data Phase Timeout Calculations

Nelson Henry Powell III
The delivery of forwarded data frames does not require a CSMA/CA. In the RX_FINAL_CH_LIST frame, the Sink announced the CHs for the current round in a particular order. This order defined the Channel Offset for the simulated environment; moreover, this order also defines the order in which the CHs will transmit their SINK_DATA frames. Each CH determines its start time based on its position within the RX_FINAL_CH_LIST frame, being 0 indexed. The position multiplied by the maximum CH time provides the start time for the SINK_DATA transmissions, as expressed in Equation 7(C).

Noting that the suggested CH:N ratio is 5%, and that the tested network is set to 100 node, the optimal LEACH round length is estimated at 2.694 seconds.
Figure 5-7: LEACH Protocol Event Timeline
5.4 LEACH Results
To validate LEACH, nodes were placed in a circular space using either an NS-3 Uniform or Random DiscPositionAllocator module. Both modules provided random layouts of nodes within the unit circle, with respect to the maximum power level transmission distance. All nodes were configured for $1\text{mJ}$ of power, and the simulation time was allowed to run for up to 500 seconds. The simulator would shut down once all nodes (except for the Sink) lost power, regardless of the time remaining.

Figure 5-8: Simulation Layout, (A) Bottom Random (B) Bottom Uniform (C) Center Random (D) Center Uniform
Two test scripts were fixed for 100 nodes plus the Sink, with the Sink located at the center of the circle for the first script, and the Sink positioned at the bottom of the circle for the second script. In order to start up the Ir-wpan PHY properly, the simulation has to run for about 500ms prior to kicking off the first LEACH round. The scripts allow for command line definition of the required number of CHs and modification of the Random Number Generator seed value. Bash scripts were used to launch multiple instances of the test scripts. Each script instance executed 50x for a given CH count, changing the PRNG seed for each run to provide for variations in the result.

Figure 5-8 presents one example image per layout. Layouts (A) and (B) present the layout wherein the Sink is positioned at the bottom of the circular space. Graph (A) demonstrates the Random layout and (B) presents the Uniform distribution. Graphs (C) and (D) illustrate the Sink located at the center of the simulated space, and (C) and (D) illustrate Random and Uniform layouts, respectively. The four graphs represent only one of 50 variations auto-generated by the modification of the PRNG seed. From the graphs, one should note that the Uniform module performs a better job of node distribution, avoiding heavy concentrations of nodes in close proximity.

It should be noted before discussing the results, that the original LEACH authors assumed that a single CH selection is comparable to a network wherein all nodes transmit their data directly to a Sink. The following sections discover sources of unnecessary power dissipation. When applied to a single CH network, the overhead involved with the use of LEACH may be worse than all nodes operating in a Direct Connection, with CSMA/CA, approach. Keeping this in mind, the proposed solution of BATSEN will revisit the Direct Connection approach to some degree.

5.4.1 Network Death Rate
Figure 5-9 does not entirely confirm the results from the original LEACH specification – the maximum lifetime of the network lies somewhere between 2% and 6% of network size dedicated to CH operations. In the graph, the first occurrence of a node death due to power loss is recorded for each CH count as the yellow dotted line. When 50% of the nodes transpire, the time is recorded as the green dash-dotted line. When all nodes die due to power loss, the time is recorded as the blue dashed line. Due to the 500ms offset to properly spin up the Ir-wpan PHY, the solid red line
is interpolated by subtracting 500ms from the last node death line, and the true maximum network lifetime is revealed.

The first observation in Figure 5-9 is that the longest time till First Node Death (FND) occurs between 7 and 9 CHs. Though the first node death is delayed, the maximum network lifetime curve is on a downward slope and converging with the First Node Death curve. As the number of CHs increases, nodes are more likely to use the Low Power transmission setting rather than transmit across the simulated space. This helps delay the initial First Node Death but increases the number of nodes awake during the Data Exchange and Sink Data phases, causing a faster death of the network.

Where the maximum lifetime occurs between 3 and 4 CHs, the third and fourth earliest FND time is seen. This is due to the distribution of CHs across the simulation space. Non-CH nodes are more frequently forced to transmit across the entire space, resorting to high power transmissions. Even with the random selection of CHs, the reliance on High Power transmissions causes a number of nodes at the periphery to drain their power faster. Since the Sink is located at the center of the simulated space, it would be more efficient for the nodes using High Power to rely on a Direct Connection transmission to save power.

Figure 5-9: LEACH Lifetime per Cluster Head count, Sink at Center, Uniform Distribution of Nodes
Another interesting aspect of LEACH is the fact that all four curves begin to converge as the number of CHs is increased. This is due to an increased number of nodes spending time with their receivers enabled during the Data and Sink Data Phases, expending power equal to the transmit power.

Figure 5-10 presents the same information for a series of tests using a random distribution of nodes in the unit circle. Again, the original LEACH hypothesis is not entirely confirmed; too few nodes causes a premature death of the network, as does too many. The curve in Figure 5-10 is shifted such that the maximum network lifetime is centered between 2% to 3% CHs.

In Figure 5-9, the mid-death curve reaches a level maximum between 4 and 5 CHs, whereas Figure 5-10 shows the mid-death curve reaches its peak at 3 CHs. The curves presented are averages over the 50 iterations per CH value. With the random position selection, it is more likely that nodes are placed closer together, or geographically clustered. With closer proximity, nodes can reduce their average power level when transmitting data to cluster heads. Net lifetime also improves with fewer CHs under the non-uniform distribution. Having more nodes in close proximity increases the number of nodes registered with a CH, which in turn increases the number of nodes sleeping per round and conserving power.
Figure 5-11 and Figure 5-12 present the results of the network lifetime when the Sink is located at the bottom of the unit circle. Close inspection reveals very minute variations in performance relative to the scenarios where the Sink is located at the center of the circle.

The Uniform node allocations across the 50 iterations are identical for both the Sink at the Center and Sink at the Bottom test scripts. Likewise, the Random node allocations across 50 iterations are also identical between the Sink at the Center and Sink at the Bottom test scripts. Therefore, comparing the pairs of graphs categorized by distribution methods demonstrates that the protocol is not affected by the location of the Sink; rather the protocol is impacted only by the distribution of the nodes and the number of required CHs per round.

Figure 5-11: LEACH Lifetime per Cluster Head count, Sink at Bottom, Uniform Distribution of Nodes
5.4.2 Number of Cluster Heads per Round
Figure 5-13 illustrates the real number of cluster heads elected per round for a given required CH setting. As described in section 5.3, the cluster head selection process is imperfect and may result in too few CHs being elected. Therefore, the sensor module was modified to record the number of CHs assigned by the Sink per round. Figure 5-13 presents the results of the uniformly distributed nodes with the Sink centered in the unit circle. Each CH line is the average across the 50 test iterations.

Of the 12 test cases, only the 2 CH setting averages out to 2 CH per round consistently. Networks configured for a single CH are erratic, typically averaging less than one CH per round. With the original LEACH cluster head selection algorithm for a single CH, nodes have less than 2% probability of self-electing for the first 30 rounds, and don’t reach a 10% probability until the 90th round. This forces the LEACH protocol to resort to the secondary CH selection process to artificially increase the probability of meeting the percent CH selection requirement.

As the percentage of network nodes required to become CHs increases, a periodic dip in the number of CHs per round begins to appear. As the number of CHs increase, the number of rounds per cycle decreases. The periodic dip follows the number of rounds required to complete a cycle.
of the protocol. This dip is explained by two issues in the waveform: reliance on a PRN for self-election, and a naturally occurring non-integer

Firstly, early in a cycle, the number of eligible nodes is large, therefore the probability of the required number of nodes self-electing increases. As the network moves forward in time, later in the cycle, there are fewer eligible nodes; however, the threshold for self-election does not increase proportionally to the number of nodes remaining. Thus, the probability of meeting the required number of CHs begins to drop.

The second issue is a result of the uneven quotient in Equation 2. When the number of required CHs does not divide evenly into the number of nodes, there is a discrepancy between the number of rounds required to complete a cycle verses the probability of self-election. If a designer chooses to use a ceiling function on the number of rounds per cycle, \( R = \lceil \frac{1}{P} \rceil \), and \( P \) does not divide evenly into 1, then there will always be one round wherein fewer than the required number of CHs are self-elected. If the implementer chooses to use a floor function, then there will always be at least one node that does not function as a CH throughout an entire cycle of the waveform.

![Figure 5-13: LEACH Cluster Heads per Round, Sink at Center, Uniform Distribution of Nodes](image)

Another major point of interest in the curve is the CH selection failure at low CH counts towards the end of the network life. When networks are configured for 2 to 8 CHs per round, the waveform
results in a large period wherein no nodes act as CHs. The 2 CH curve is most pronounced where the number of self-elected CHs begins to drop off around round 90 and does not recover until around the 105th round. As the number of configured CHs increases, the curve is less severe, but is still noticeable.

The loss of CHs is attributed to the death of nodes in the system. As time progresses, nodes begin to lose so much energy that they may not complete a full round. The dip in CH count occurs as there are rounds that occur where the nodes that have not participated in the network as a CH for a given cycle cannot self-elect as they are out of power. To recover, the network must traverse the number of rounds remaining in the current cycle and start a new cycle with all eligible nodes resetting their threshold calculations.

This process increases the rate of node death in that all nodes resort to Direct Connection status with the Sink. During rounds when there are no nodes with power left to act as CHs, the nodes with power must actively listen through the first 3 phases of operation to detect the lack of CHs. The active listening causes power drain at a similar rate to the transmission power loss. Once the Sink announces the transition into the Data Phase, all remaining nodes attempt to send their data directly to the sink using CDMA/CA. The only power saved is the sleep time between the transition into the Data Transmission Phase and their transmission attempt, and the time between their transmission completion and the start of the next round.

The higher-level CH counts seem immune to this affect, but this is not true. The smoothness of the higher CH counts is due to the initial system power settings. The high CH counts consume power faster than the lower CH counts, and in so doing, reach a network end of life before the lower CH counts. As an example, looking closer to the 2 CH configuration, a complete cycle takes 50 rounds. With the initial power setting, nodes begin to die in their second cycle, at approximately round 35, or round 85 for the graph. This allows for 15 rounds of all nodes operating as Direct Connect nodes before the next cycle begins. Once the next cycle begins, the few remaining nodes start functioning as CHs until all node are dead.
However, for a large CH count network, the network just happens to die before the next rounds begins. If the system power is increased, the same loss of CHs would be visible at higher node counts. Looking at the curve for 9 CHs, the final drop in active cluster heads begins around round 75. Prior to the drop, there is a small drop in the curve near round 70; note that there are 12 rounds per cycle for 9 CH networks. The drop rapidly recovers at round 72, which is the beginning of the next cycle. Before the curve can recover completely, the power loss is ubiquitous and the network dies by the end of the next cycle.

![Figure 5-14: LEACH Cluster Heads per Round, Sink at Center, Random Distribution of Nodes](image)

Comparing Figure 5-13 to Figure 5-14, uniformly distributed networks versus randomly distributed networks, one can see the similarities in the actual versus programmed CH count curves. However, the notable difference between the curves is the number of function rounds for lower CH counts. The distribution of the nodes has the biggest impact on the waveform’s behavior. Larger CH counts seem to result in a similar number of active rounds, which is attributed to the increased probability for geographical clustering.

Another major difference in the two curves is the loss and recovery of self-elected CHs in the later rounds. As the networks in the random layouts approach the end of their second cycle, fewer nodes have died due to the reduction in power dissipation. Therefore, the number of rounds without self-elected CHs is reduced, reducing the power expenditure in the Direction Connection mode. This compounds and results in the extension of the network’s lifetime.

Nelson Henry Powell III
Figure 5-15 and Figure 5-16 illustrate the effects of the Sink position for the same Uniform and Random distributions. Comparing Figure 5-15 to Figure 5-13, it is apparent that the Sink’s position has little impact on the function of the network. There are variations in the individual curves, but nothing significant as to draw a conclusion based on the sink’s position. Similarly, the same can be said for a comparison between Figure 5-14 and Figure 5-16.
5.4.3 LEACH Packet Rate per Round

Figure 5-17 and Figure 5-18 presents the number of packets per second, per round. Once again, the single CH setting is most erratic, with no consistent packet rate and early death of the network. This is due to the inconsistent CH selection demonstrated in Figure 5-13 and Figure 5-14, coupled with the heavy power drain using the Direct Connection methodology.

Regardless of the network layout, the packet rates for CH configurations above 1 CH are consistent. Note that the 2 CH configuration produces the highest actual packet rate; however, the 3 CH setting is the second highest packet rate and the only stable packet rate unlike the 2 CH setting. Furthermore, the CH settings from 2 to 12 provide similar packet rates within 15% of each other.

The packet rate is calculated by dividing the number of packets that reach the sink every round by the round length in seconds. The theoretical maximum data rate for a LEACH network, using 100 nodes, ranges from 41.1pkt/s for a 1 CH network to 38.1pkts/s for a 12 CH network. However, knowing that two Advertisement phases are normally required for correct functionality, the theoretical maximums are reduced to 40.4pkts/s and 31.9pkts/s respectively. These data rates are supported in that the maximum difference in round length, between 1 CH and 12 CHs, is 654ms.

The similar round lengths is a result of the Scheduling Phases inverted time relationship. Equation 5 shows that the Scheduling phase gets shorter as more CHs are required. This allows large CH counts to reduce much of their round time. The other phase calculations use the number of Clusters Heads as an additive property, causing the slight reduction in throughput. Elimination of one Advertisement phase and reworking the Data Phase algorithm may allow for more consistent throughput rates as well as power consumption rates.
Again, Figure 5-17 and Figure 5-18 accentuates the different between layout distributions. Higher CH counts live for similar round counts, whereas the lower CH counts with Random distributions live longer than networks with low CH counts and Uniform distributions. This is reiterated in Figure 5-19 and Figure 5-20, where the Sink is moved to the bottom of the test circle.
5.5 The Real LEACH
The authors of LEACH presented the eight features of the protocol as listed in section 4. Of the features, 50% of the features are either unrealistic, unachievable, or introduces tradeoffs to achieve the feature. It was shown that achieving a consistent CH assignment is unrealistic with the original random number selection algorithm. Adding the Secondary Advertisement phase improved the results, but limitations in the distribution of CHs per round forces a non-constant function.
Moreover, CH self-election is not necessarily self-actuated. In times when too many nodes self-elect, the Sink is forced to resolve the situation by transmitting the final list of CHs, reducing the extra CHs to standard member nodes. Therefore, the Self-Elected CH feature is not really a feature, as the waveform must shift to a centralized control methodology to ensure correct operation.

The Reduced Frame Contention through TDMA with DSSS only applies to the Data Phase operations. During the Advertisement, Secondary Advertisement, and Setup phases, nodes resort to CDMA/CA to perform the necessary network management to achieve the desired savings. With the CDMA/CA, there is a tradeoff between successful network management and power savings. In this experiment, the implemented protocol accepted the power loss in order to stay true to the original protocol description.

In addition to the heavy use of CDMA/CA, DSSS is not the panacea to channel contention as assumed by the authors. During the Data Transmission Phase, all member nodes are scheduled for channel access in the same time slots for each CH. Since the network is synchronized in time, the usable diameter of any cluster is reduced for every additional transmitter in the same time slot. The diameter shrinks when two or more transmitters with different spread codes, different path loss due to distance, and potentially different transmit power levels have unequal power at the receivers. With DSSS, this results in destructive interference at each receiver. If the interference is enough to lower the SINR below the sensitivity of the receiver, data is lost and energy is wasted.

It is not clear as to why the reduced network diameter was considered a feature of the original LEACH waveform. Reducing the network size does present the possibility of clustering nodes physically to reduce the required transmit power, but as a network gets too small, it eliminates any advantage the Adaptive Power Level feature provided.

Finally, the notion that LEACH provides “complete distributed control of the network” is not entirely true. When placed into a more realistic simulation environment, where the simulation engine does not impose omniscient knowledge or “behind the scenes” synchronization upon every
node, the waveform must take on certain implementation tradeoffs. In order to remain true to the original specification, the waveform must shift to more centralized control. If the implementer is satisfied with an unstable CH count, resulting in high quantities of Direct Connection nodes and excessive power loss, then keeping with the decentralized design can be maintained. If power conservation is a high priority, then the centralized control needs to be added.
6 The BATMAN-SENSOR Protocol

The realistic implementation of LEACH revealed a number of issues that must be surmounted in order to optimize the power dissipation. Section 3 revealed that BATMAN was not optimal for all scenarios, but still has some potential. Therefore, BATMAN was examined for features and behaviors in order to design and develop a BATMAN variant known as BATSEN.

6.1 LEACH as a Starting Point

Though the LEACH protocol did not perform exactly as expected, one cannot relegate the revolutionary step towards power optimization introduced with the CH concept. Besides the notion of CHs, there are a number of features that idealistically enhance a waveform’s operation. Therefore, before creating a BATMAN for sensor networks, it is important to identify desirable features and practical assumptions that could be leveraged.

6.1.1 LEACH Protocol Requirements

The LEACH protocol advertised a number of features as presented in section 5.2. Of the features described by LEACH, the BATMAN derivative protocol will attempt to achieve the following six features, assuming modifications to the features:

- Decentralized Forwarder elections
- Adaptive Transmit Power
- Rotation of Forwarder responsibilities
- In-network processing (compression) of aggregate data
- Scheduled sleep cycles
- Distributed control of the network

It was shown in LEACH that nodes could not reliably self-elect as CHs and expect to achieve the nominal 5% of the population as CHs ratio. At some level, centralized control or additional controls needed to be added to support optimal selection. Additional controls lead to the increase in the Setup-Phase, causing excessive power loss. Therefore, the BATMAN protocol approach
will look to use a non-Sink election method, one wherein nodes may self-elect or peers may elect the appropriate CH for their given location.

The power management, or Adaptive Transmit Tower, is considered a standard feature of sensor nodes. If the sensor node has this ability when deployed with LEACH, the option should be present in the BATMAN approach as well. There is no change to this feature requirement.

The random rotation of CHs was mathematically difficult to achieve without intervention. In keeping with the non-Sink approach to CH election, rotation of CHs may not rotate perfectly. Therefore, the feature is downgraded from Random Rotation of Forwarders to just Rotating Forwarders. This feature provides leeway for the BATMAN approach to reuse a Cluster Head-like concept if the situation deems the reuse is optimal.

The BATSEN protocol will assume that in-network processing is used to reduce the data traffic based on the close proximity of nodes to their selected forwarders. The LEACH implementation relied on the CH forwarded data payload containing member node identifiers and frame sequence numbers to properly track the flow of data frames. This method will be used in the BATMAN protocol for fairness in the comparison.

LEACH also professed complete distributed control of a network. Sleep cycles must be allowed whenever possible to avoid the waste of energy. Therefore, a synchronization method is required that can reduce or eliminate the complete distributed control of the network. BATMAN will be modified to support sleeping when possible, but also acknowledges that there must be some method of network synchronization or centralized control to enforce sleep patterns without the loss of data.

Staying true to the original BATMAN protocol, the proposed network must rely on CDMA/CA, and thus the TDMA assignment strategy of LEACH is ignored. It was also shown that the reduced diameter of the network was not so much a feature as it was a test strategy that impacted performance. Neither of these two original features are included in the proposed BATMAN solution.

Nelson Henry Powell III
Ideally, the BATMAN approach should seek to minimize active periods and maximize sleeping period while maintaining control of the network. By implementing a small active to sleep ratio, energy is preserved. At the same time, the active to sleep ratio must also fit in a period of time similar to that of the LEACH FSM round length. This will allow for a direct comparison between the two protocols.

6.1.2 LEACH Environment Assumptions
In the original LEACH paper and supporting NS-2 software, there were a number of explicit and implicit assumptions made by the authors. Of the eight identified assumptions, only four would carry forward into the BATMAN protocol.

1) No mobility in the system
2) Node count known a priori
3) Geography of the network is always less than the radius of a transmitter’s capability
4) Topology is uniformly distributed

To directly compare LEACH and BATMAN, mobility must not be a factor. Moreover, the NS-3 test scripts are designed to repeat the same layout per test iteration. This allows BATMAN and LEACH to execute two separate script instances against the same random layout. Touching on the fourth assumption, the NS-3 environment will test both uniformly distributed networks and purely random distributed networks.

Node count is always considered a configuration item. It is a design objective that the BATMAN protocol not require a preconfigured node count, and thus implement a method of neighborhood discovery. By adding a Late Net Entry (LNE) capability, dead nodes may be replaced which is not a feature of LEACH.

In keeping with the LEACH protocol, the boundaries of the simulated test areas are kept at the 1% PER for high power (-1dBm) transmissions. Thus, all tests continue to rely on a circular network to ensure the use of the max power is possible. Furthermore, the test environment relies on the use of both Random and Uniform allocation models to seed the distributed sensor topology.

Nelson Henry Powell III
Unlike the preceding four assumptions, the next set of assumptions are eliminated from the BATMAN approach. Since BATSEN is designed using more holistic methods, assumptions based on the limitation of a simulation environment are considered invalid and must not be used for design decisions.

1) All links are always bidirectional and symmetric
2) RSSI accurately represents distance
3) Spread spectrum guarantees no interference during the Data Transmission phase
4) All nodes make successful decisions

First, it is unrealistic to assume that all links are symmetric. In wireless communications, to assume a bi-direction, symmetric link is to assume failure in MAC design. Wireless links are inherently lossy; there are various channel affects, such as fading, doppler, and destructive interference, that negatively impact the probability of success of frame reception. If hardware does not support a SINR feature, it is difficult to evaluate a link. MIT introduced the ETX concept, the inverse of a Delivery Ratio multiplied by the Reverse Delivery Ratio, as a method of evaluating link quality which is still used today in modern MANET routing algorithms.

Similar to the link symmetry, RSSIs are not as reliable as one might think. RSSI is a measure of received power at the time of frame reception. As discussed in section 5.2.7, the use of DSSS can cause a serious degradation of the channel when power levels are not balanced at every receiver. If the channel is shared using DSSS, then the RSSI will increase for every active cluster in a network, as each TDMA slot is almost guaranteed to have an orthogonal (due to different spread codes) transmitter occupying the same slot. To this end, a SINR provides more insight as to the strength of a signal when gauging the distance to a peer.

The final assumption, that all nodes make successful decisions, is based on the invalid omniscience within the test environment. Unless there is a wireless or wired data exchange, no two nodes may know information about the node’s decisions or perception of the network. Additionally, no two nodes can be synchronized without a timing exchange or outside timing source. The original LEACH software made such assumptions, allowing for a theoretical maximum in the network’s
performance. For the purposes of this paper, omniscience is eliminated; all synchronization must either be explicitly synchronized or centrally controlled from the Sink.

6.2 BATMAN Inheritance

BATMAN routing is designed to operate on a wideband, low latency, lossy, wireless network (typically IEEE 802.11). BATMAN has many advanced features to support routing within a mesh as well as supporting access to exterior networks, known as Host Network Associations (HNA). Traffic in a BATMAN mesh is generated based on individual node functions and may not require gateway access.

BATMAN supports the following features set in the version IV daemon:

1) Periodic Originator Message (OGM) transmission with jitter
2) Minimized OGM frame format
3) Transmission Quality assessment
4) Link Symmetry detection
5) HNA announcement
6) OGM Aggregation for reduce overhead
7) Best Next-Hop Neighbor ranking
8) Elimination of complete topology exchanges

With traditional MANET routing, nodes send data to one another using Unicast, Broadcast, or Multicast, as well as to external networks connected to Gateway nodes. Unlike BATMAN, sensor networks are concerned with the delivery of sensor data to a centralized Sink node for post-processing and analysis. This data delivery model is known as Convergecast. In spite of the differences between Sensor Networks and MANETs, both networks desire a Best Path to a destination with minimal expenditure of energy. Therefore, the HNA announcement is rendered useless for sensor networks. However, the Sink is analogous to a centralized gateway.

Developing a BATSEN protocol meant starting with the OGM message concept. In BATMAN, every node transmits an OGM describing who it is, what HNAs are associated with the node, and initial values for the source Transmit Quality (TQ) and Time To Live (TTL) fields. As OGMs are

Nelson Henry Powell III
received from peer nodes, the TQ values for each peer are calculated and the OGMs are stored for future retransmission. To avoid excessive fragmentation of the channel, nodes make a best effort to aggregate received OGMs and their self-source OGM into a single frame.

OGMs are also tracked based on sequence numbers and a sliding window algorithm. Reception of an OGM authenticated as being new, allows a node to shift a bit vector signifying a nodes historical presence and stability. This feature is critical in that BATMAN is primarily a Link-State routing protocol. OGMs will not support a TQ, rather the BATSEN OGM will rely on exchanging SINR values to detect distance and channel conditions between node pairs.

As the basis for BATSEN, the protocol begins with a Sink sourced OGM. The Sink controls the cyclical nature of the network by periodically transmitting an OGM notifying one-hop neighbors of an epoch boundary. The OGM is assumed to be fixed to a precise period, typically aligned to Time of Day (ToD) using a Network Time Protocol (NTP) when attached to a network or a Global Positioning System (GPS) 1-Pulse Per Second (1PPS) signal. The Sink is the only node to source the periodic OGM.

Member nodes also transmit OGMs, but only after receiving the synchronization OGM from the Sink. Unlike the Sink, member nodes attempt to resolve a random transmit time between the received Sink OGM and the predicted next Sink OGM reception. The member nodes also add a pseudo-random jitter offset in an attempt to reduce collisions and detect a busy channel. This effectively recreates the first feature of BATMAN.

The first time an OGM is transmitted, it provides peer nodes a notion of the sender’s presence. Once the first OGM is sent, succeeding OGMs include information about 1-hop peer nodes and the Sink. The information encapsulated should provide information regarding a node’s link quality to peers as well as the Sink. Since member nodes are not gateways to other networks, there is no need for an HNA field in the OGM. This implements the third feature of BATMAN and eliminates the fifth feature, as there are no HNAs.
Much like BATMAN, BATSEN will also de-centralize the “routing” schema. After OGMs are exchanged, nodes should attempt to elect an optimized set of Forwarders. Here, Forwarders in BATSEN are analogous to CHs in LEACH. Rather than nodes self-electing, nodes will use more of a swarm mentality, using information from received peer OGM messages to derive a similar solution set without exchanging additional messages.

Unlike LEACH, BATSEN shall not mandate that nodes send all sensor data messages to a Forwarder; rather, as with a standard routing protocol, BATSEN will allow nodes close to the Sink to resort to Direction Connections to avoid unnecessary transmissions. This single point indicates a need to resolve a link’s minimum required transmit power to determine the best path to the Sink. This will impact the notion of Transmission Quality.

Unlike BATMAN over WiFi, there is not a lot of bandwidth or energy to waste on routing overhead. Therefore, using the ETX or Echo TQ values, as discussed in section 2.1, would be over-burdensome. Both methods require multiple transmissions and tracking of receptions to determine a TQ or ETX factor before nodes begin to exchange data. Based on this, and the aforementioned minimum transmit power requirement, the BATSEN protocol replaces the TQ value with a Received SINR value. The use of SINR provides an alternate approach to implementing both the third and fourth features of BATMAN.

IEEE 802.15.4 uses 127-byte frames. Therefore, the information passed in a single OGM must be kept at a minimum. If the OGMs are much smaller than the 127-byte limit, received OGM information can be inserted into future node sourced OGMs. Rather than repeating a complete OGM, information from received OGMs can be stripped down to only essential data elements and added as Type, Length, Vector (TLV) fields in the node’s source OGM. This allows for optimized OGM aggregation, the sixth feature of BATMAN.

Using the SINR information about links to peers in conjunction with received OGM SINR data, each node can resolve a connectivity graph. Since there is no movement for this setup, there is no immediate need for a timeout or history vector similar to the one found in BATMAN. The connectivity graph is broken down into lists of source nodes at a given power level. For the Sensor

Nelson Henry Powell III
PHY, there are three power levels, hence there are three lists of peer nodes. Each node in the list contains an internal connectivity graph based on the last OGM received from the associated node. For every peer OGM received, a data structure representing the node is added to one of the power level lists or is updated if it already exists. The information about the node and its peers is used to calculate a relative score. The local node will also use the same formula against itself. The node with the best score is considered the best candidate for a Forwarder. Node then exchange their best candidate information in an attempt to resolve a set of best candidates for the network. This process implements the seventh feature in BATMAN.

6.3 BATSEN Frame Formats
Before describing the protocol, the BATSEN frame formats are provided. BATSEN, operating on an IEEE 802.15.4 PHY, uses 16-bit MAC addresses for a maximum network size of 65535 nodes not counting the Sink. Similar to traditional IEEE headers, the Destination MAC proceeds the Source MAC, followed by an 8-bit Type field. Since the payload of a frame may vary, an 8-bit Length field follows the Type; the maximum length of an 802.15.4 frame being 127-bytes, the Most Significant Bit of the length field should always be 0.

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Frame</td>
<td>00</td>
<td>OGM Transmitted by Sink only</td>
<td>Figure 6-2</td>
</tr>
<tr>
<td>NULL</td>
<td>01</td>
<td>Presence notification OGM</td>
<td>Figure 6-3</td>
</tr>
<tr>
<td>RX Power</td>
<td>02</td>
<td>Listing of peer nodes by power group</td>
<td>Figure 6-4</td>
</tr>
<tr>
<td>Router Select</td>
<td>03</td>
<td>Next-Hop election frame</td>
<td>Figure 6-5</td>
</tr>
<tr>
<td>Data Frame</td>
<td>04</td>
<td>Node to Next-Hop or Node to Sink data frame</td>
<td>Figure 6-6</td>
</tr>
<tr>
<td>Forward Frame</td>
<td>05</td>
<td>Next-Hop to Sink data frame</td>
<td>Figure 6-7</td>
</tr>
</tbody>
</table>

The next field is a 32-bit Sequence number, unique to the source of the frame. At the end of the frame is a 16-bit frame Checksum, use for validation. The algorithm used for the checksum is left to the implementer and is out of scope of this document. Between the Sequence Number and the Checksum is the payload specific to the message. Payload lengths range from 0 to 115 bytes.
The first frame type is the Super Frame, named aptly after the Super Frame from the IEEE 802.15.4 specification. The Super Frame follows the format illustrated in Figure 6-2. Every Super Frame presents the Round number as a 16-bit unsigned integer. This is used to synchronize member nodes to the network. The Flags field is a 16-bit field providing specific instructions to member nodes. Presently there are only three Flag types, but the field uses 16-bits to ensure the hardware uses 16-bit registers for code and hardware optimizations. CPU memory alignment is outside of the scope of this document, but awareness of alignment requirements is the impetus for maintaining word alignments in message formats.

The final two fields in the Super Frame are two 16-bit fields: the number of active nodes in the network and the number of 1-hop forwarders required for network optimization. The Number of Nodes (NoN) field is a count of nodes that the Sink has heard from, each having sufficient power level for operations, and can be counted on to function as a node in the current round. The Number of Forwarders (NoF) field is the maximum number of forwarders allowed for the current round based on the NoN multiplied by the configured percentage of nodes to act as forwarders.

Non-Sink OGMs use the standard BATSEN frame format, as illustrated in Figure 6-1 and a sub-header as presented in Figure 6-3. The base message includes an 8-bit field for the current transmit power level and an 8-bit field for the percentage of power remaining for the source node. As with the LEACH protocol, there are only three power levels (-1dBm, -11dBm, -19dBm) requiring three unique 8-bit patterns to represent the active transmit power. Multiplying the power percentage by 255 scales the percentage value to the 8-bit field. The two fields together maintain the 16-bit word width for code portability.
The NULL OGM is a presence message generated by all non-Sink nodes. The OGM does not provide any additional data than the base Non-Sink OGM format in Figure 6-3. Nodes shall always transmit NULL OGMs using the highest power available to the node. By transmitting a NULL OGM, receiving nodes record both the presence of the source as well as the minimum power for the source to reach the receiving node.

A node uses the RX POWER OGM to report each peer node’s minimum power level required to reach the source node. The payload follows the standard non-Sink OGM format. Transmitting the RX POWER OGM at High power ensures maximal coverage of the Power message and attempts to compensate for missed NULL OGMs due to collisions.

After the base NULL OGM fields, a Power Level header and associated address data set are appended. The Power Level header is composed of a 1-byte TX Power Level flag followed by a 1-byte count of addresses that follow the Power Header. Each address is a 16-bit address, allowing for a maximum of 53 addresses in a single RX POWER OGM. If the network size is too large to fit all nodes in a single RX POWER OGM, then multiple OGMs should be used.

As illustrated in Figure 6-5, the RTSEL OGM, or Router Selection OGM, provides the election delivery mechanism allowing nodes to broadcast the address of their desired Next-Hop address.
The OGM uses the base OGM format, followed by the RTSEL TLV and a length field with a constant value of 0x01 hex. The address of the desired Next-Hop MAC address follows the TLV header.

The RTSEL OGM serves a secondary purpose in the BATSEN protocol, that of a forcing function for the LNE operations. When a local node has indirect knowledge of a new neighbor, the local node will add the neighbor to a Not Found list. Indirect knowledge may occur when the local node fails to receive any OGMs from the peer due to collisions, or the node is a new node recently added by the network administrator.

The list is appended to the next RTSEL OGM available in the form of a TLV. The additional TLV, known as the NOT_FOUND message, is appended to the RTSEL payload as another 8-bit message type, an 8-bit address count, followed by the list of 16-bit MAC addresses. Reception of the TLV signals the Sink to force a RX Power OGM exchange.

![Figure 6-5: BATSEN Router Selection OGM Frame Format](image)

Though the frame header is different, the Data Frame used for BATSEN, as seen in Figure 6-6, serves an identical purpose as the Data Frame used in the LEACH implementation. The sequence number in the base header supports a disparate purpose from the globally unique sequence number in the payload. In a real implementation, there would be no globally unique sequence number; it is merely an artifact used for the validation of the protocol.

In total, the packet is fictitious and meant to represent a real 100 Bytes sensor payload. Data Frames may be sent to Next-Hop nodes or directly to the Sink. Nodes that receive a Data Frame destined for a different node should not aggregate or retransmit the information if they are a forwarder. In a real-world implementation, a Sink may store data directed at a Forwarder; for this environment, a Sink will ignore frames not directed at the Sink or the broadcast MAC address.
The Forwarder Frame format is again similar to the Sink Data frame in the LEACH protocol specification. Each Forwarding station must aggregate all MAC addresses and associated globally unique sequence numbers collected during the Data Exchange phase and transmit a Forwarder Frame to the Sink.

6.4 The BATSEN Finite State Machine

For this project, there is a base assumption, much like LEACH, that the BATSEN protocol uses a prepopulated network node count. This allows for software to pre-calculate any timer values needed for state transitions. It should be noted that unlike LEACH, preconfiguring the node count is temporary, as it will be shown later on that the BATSEN protocol may dynamically add nodes to the network count as a form of Late Net Entry (LNE).

The second assumption for the FSM is that all transmissions from the Sink are performed at High power. This ensures the Sink can reach all nodes in the network regardless of their location within the simulated boundary. The Sink is considered tied to an infinite power source and does not count against the network lifetime.
The BATSEN protocol is broken into four major states: Startup, Power Collection, Router Selection, and Data Exchange. From the perspective of the Sink, the Router Selection and Data Exchange phases are synonymous, and are known as the Periodic OGM phase.

6.4.1 Startup Phase
The first phase in the FSM (Startup) provides an early node presence detection mechanism and is not part of the fundamental portion of the FSM. When powered on, a node listens for a Super Frame from the Sink; nodes expect to receive the NULL OGM Super Frame. Receiving the NULL OGM Super Frame causes the node to enter the Startup state, as illustrated as transition 1 in Figure 6-8.

The NULL OGM Super Frame instructs all receiving nodes that they must transmit a NULL OGM to establish network presence. Every node attempts to schedule a NULL OGM during the Null-OGM Period. The time-period for nodes, defined by Equation 8(A), is the number of nodes (N) times the length of a frame transmission $T_{SLOT}$. Since BATSEN uses CSMA/CA, a constant
multiplier \((k_1)\), that is greater than 1.0, is used to provide additional time to reduce the probability of collision. For the simulation implementation, the Null Period constant is 1.25.

\[
T_{\text{NULL}} = \left[ T_{\text{SLOT}} \times N \times k_1 \right]
\]  
\(\text{(A)}\)

\[
S_T \sim \text{U}\left[0, \left\lfloor \frac{T_{\text{NULL}}}{T_{\text{SLOT}}} \right\rfloor \right] : \quad S_T \in \mathbb{Z}
\]  
\(\text{(B)}\)

\[
r \sim \text{U}[0,1] : \quad r \in \mathbb{R}
\]  
\(\text{(C)}\)

\[
T_{\text{XMIT}} = (T_{\text{SLOT}} \times S_T) + \left[ (T_{\text{jitter}} \times r) - \left( \frac{T_{\text{jitter}}}{2} \right) \right]
\]  
\(\text{(D)}\)

\[
T_{\text{NULL}} = [T_{\text{SLOT}} \times N \times k_1] + T_{\text{SLOT}}
\]  
\(\text{(E)}\)

Equation 8- Null Time-Period

To reduce the probability of collision, BATSEN uses a pseudo-random TDMA calculated by equations 3(B) through 3(D). The Null period \((T_{\text{NULL}})\) is divided equally by the maximum transmit slot length \((T_{\text{SLOT}} \text{ or } 7\text{ms})\). A transmit slot \((S_T)\) is then randomly picked by each node, where \(S_T\) is a random natural number between 0 and \(\left\lfloor \frac{T_{\text{NULL}}}{T_{\text{SLOT}}} \right\rfloor\). Due to the \(K\) multiplier in Equation 8(A), there are more slots than nodes, thus reducing the probability of collisions.

To further reduce the probability of collisions, BATSEN utilizes jitter just like BATMAN. Since the Super Frame arrived at each node at a different time, due to propagation delay, the slots are not lined up perfectly. But the propagation delay is not a significant source of jitter to avoid collisions by equidistant peers. Therefore, a jitter constant \((T_{\text{jitter}})\) is multiplied by the pseudo-random number \(r\). The random number \(r\) is selected from the set of all real numbers, with uniform distribution, between 0 and 1. Next, the resulting random offset is reduced by one-half of \(T_{\text{jitter}}\), effecting a random +/- offset to the start of the slot time.

It should be noted that selection of the 0\(^{th}\) slot results in an immediate transmit attempt. To avoid negative time offsets from the current time of day, software must detect the selection of the 0\(^{th}\) slot, and skip the subtraction of \(\frac{1}{2} T_{\text{jitter}}\) during the jitter calculations. The node simply queues the OGM for transmission and begins the Channel Assessment function of the CSMA/CA media access controls.
For synchronization purposes, the Sink uses a similar timeout period, except that its timer is started prior to transmitting the Null Super Frame. Therefore, the Sink’s Null Time Period must account for the single Super Frame transmission as calculated in Equation 8(E). Since the Super Frame is much less than 127-bytes, the frame does not take up the entire $T_{\text{SLOT}}$. The additional time must also account for propagation delay. Considering the radius of the high-power transmission is only 100m, and that the speed of light is approximately 300,000km/s, the maximum propagation delay is around 300ns. Therefore, the 7ms slot time is sufficient to account for both the propagation delay and the transmission of the Super Frame.

All nodes transmit their NULL OGMs using High power. This ensures that all transmissions are able to reach the furthest location in the simulated space. The High-power transmission also allows for a link quality assessment. Nodes that receive peer NULL OGMs follow transition 6 in Figure 6-8. Receivers record the source MAC address and the SINR of the reception. The SINR is evaluated to determine the minimum transmit power required by the source to reach the receiver.

When receiving a High-power transmission, there is a set of minimum SINR values of which each correlate to a 1% PER at a lower power level. Using the PER curves in Figure 4-3 and empirical testing on a noise free channel, a high power SINR value is determine for both the Medium and Low-power settings. A node receiving a NULL OGM uses the preprogrammed SINR estimations as a method of analyzing the link between the receiver and the source. The receive adds the source node MAC address to the data structure containing all source nodes for the given power level.

At the end of the Startup phase, the Sink will transmit one or more RX POWER OGM frames, providing the transmit power required by any member node to reach the Sink. Nodes receiving the Sink’s OGM follow transition 7. Nodes that receive the OGM from the Sink, whose MAC address is in the OGM list, store their minimum power level in a data structure for future comparisons.

Nodes also store information about peer nodes, and their minimum transmit power required to reach the Sink. This information helps nodes determine the relative distance from Sink to Node and is used in election calculations later in the FSM. If a node has not heard from one of the nodes...
listed in the Sink’s OGM, the listed node is added to a separate list of nodes that are considered present, but for which no bi-directional link has been established. This list provides the impetus for the NOT_FOUND TLV used in the RTSEL OGM.

This phase is an enhancement to the core FSM used to reduce traffic later in the protocol’s lifecycle. This operation attempts to maximize the number of received MAC-SINR entries in every node’s database. By recording presence and SINR before the Power Collection phase, the protocol may be able to reduce the number of NOT_FOUND TLVs transmitted during the Router Selection phase. The advantage of early presence notification is explained in section 6.4.4.

6.4.2 Power Collection Phase
The Power Collection phase is the first of the three fundamental FSM states. From this state forward, all states provide the fundamental utility of the protocol. The Power Collection phase begins with the Sink transmitting the RX POWER Super Frame. This message immediately follows the Sink’s RX POWER OGM transmitted at the end of the Startup phase. Receiving the Super Frame forces all nodes to follow transition 3, as illustrated in Figure 6-4, and create RX POWER OGM frames listing peer MAC addresses for each of the available power levels.

As an optimization note, frames are always created and transmitted starting with the Lower-Power lists working up to High-Power lists. As frames are transmitted using CSMA/CA, nodes will back-off when the channel is occupied by another transmitter. As the Power Collection phase nears the end of the timeout, it is possible that some frames do not get transmitted. Therefore, programmatically, nodes always assume the need to transmit using High-Power to reach all other peers until they receive an RX POWER OGM that instructs them otherwise. Therefore, the effects of lost frames at the end of the phase are greatly reduced.

For each RX POWER OGM received, following transition 8 in Figure 6-4, all nodes (including the Sink) update their database regarding each node in the frame’s list. For each node in the list, the relative database entry is modified by adding the Source of the frame to a power-level list for the listed node. This allows a node to determine the number of nodes at each power level a neighbor node can reach. This information is used later in the router selection stage.
\[ T_{RXPWR} = [(k_2 \times N) \times (M \times T_{SLOT})] \]  
\[ S_T \sim \bigcup \left\{ 0 \left[ \frac{T_{RXPWR}}{T_{SLOT}} \right] : S_T \in \mathbb{Z} \right\} \] 
\[ T_{RXPWR} = [(k_2 \times N) \times (M \times T_{SLOT})] + [T_{SLOT} \times M] + (2 \times T_{SLOT}) \]  

Equation 9- RX Power Time-Period

All nodes must transmit their OGM within a limited time window, \( T_{RXPWR} \). As in the Startup Phase, nodes expect the Sink to use a similar time period for transition to the Router Selection phase, and thus must transmit their OGM(s) prior to the Sink’s next Super Frame. Using Equation 9(A), nodes determine the maximum time for the Power Collection phase (\( T_{RXPWR} \)). Again, nodes use a pseudo-TDMA channel access by subdividing the \( T_{RXPWR} \) period into discrete transmission slots, and randomly select a slot as seen in Equation 9(B). Equations 3(C) and 3(D) are reused to apply a random jitter offset to the final transmit time (\( T_{XMIT} \)).

Unlike the Startup phase, transmitting RX POWER OGMs for large network sizes may require more than one OGM per node. Following the format provided in Figure 6-3, the Base OGM has 14 bytes of overhead, leaving 113-bytes for the Power Level TLV (2-bytes) and MAC addresses (2-bytes each). Therefore, only 55 addresses can be packaged in a RX POWER OGM with a single power level list. If all three power levels are listed, the frame can contain a maximum of 53 addresses.

The \( T_{SLOT} \) time must be multiplied by the number of frames required to package all peer nodes in OGMs. Fortunately, the number of nodes in the network is transmitted in every Super Frame. All nodes can calculate the frame multiplier \( M \) by dividing the number of nodes supplied by the Sink by 53 nodes per frame. Therefore, the multiplier \( M \) is added to Equation 9(A) to support larger networks.

Since the hybrid-TDMA is decentralized, all nodes randomly select a slot in which to transmit, the probability of collision still exists. If too many collisions occur, then router selection becomes sub-optimal. Therefore, a constant multiplier \( k_2 \) is used to increase the time of the Power Collection phase. This reduces but does not eliminate collisions. Theoretically, the greater the

Nelson Henry Powell III
success earlier on in the networks lifetime, the fewer times the Sink will return to the Power Collection phase, and thus reduce the overall length of the average round.

Note that the Sink uses Equation 9(C), as its timer starts prior to the Sink’s transmission of the RX POWER OGM(s) at the end of the Startup Phase, not at the RX Power Super Frame. To maintain synchronization, the Sink must account for the number of RX POWER OGMs it must transmit. The Sink must also account for the RX POWER Super Frame transmission and add an additional transmission slot to ensure the Sink’s timer expires after the nodes’ timers expire.

6.4.3 Router Selection Phase
The Router Selection phase is marked by the Sink transmitting the Periodic-OGM Super Frame; this is the second core state in the FSM. Upon receiving the Periodic Super Frame, all nodes must evaluate their database for their best Next-Hop node. Evaluation requires that nodes complete the scoring algorithm, presented in Equation 10, for every known neighbor in their database as well as for their own. The node in the database with the highest score is the desired Next-Hop node for the local node.

$$\text{SCORE}_N = \left[ \frac{\text{CNT}_\text{SCORE} + \text{SINK}_\text{SCORE} + \% \text{Power}}{3} \right]$$

Equation 10- Node Scoring Calculation

There are three components of the dividend: aggregate node count score, the relative power to sink score, and the power remaining percentage. Each component is a real number between 0 and 1. Dividing by three produces a real number between 0 and 1.

$$S_{\text{LOW}} = 100 \times \# \text{LPP} \quad (A)$$
$$S_{\text{MED}} = 75 \times \# \text{MPP} \quad (B)$$
$$S_{\text{HI}} = 65 \times \# \text{HPP} \quad (C)$$
$$\text{CNT} = \# \text{LPP} + \# \text{MPP} + \# \text{HPP} \quad (D)$$
$$\text{CNT}_\text{SCORE} = \left[ \frac{S_{\text{LOW}} + S_{\text{MED}} + S_{\text{HI}}}{100 \times \text{CNT}} \right] \times \frac{\text{CNT}}{N} \quad (E)$$

Equation 11- Aggregated Node Count Calculation
The method of calculating the aggregated node count score is illustrated in Equation 11. A node determines the number of Low-Power Peers (LPP) and multiplies the count by 100. In a similar fashion, the Medium-Power Peers (MPP) and High-Power Peers (HPP) are counted and multiplied by a lesser ratio, respectively. This lends the node count score to favor nodes with high numbers of LPPs over nodes with higher HPPs. Next, a total count of neighbors (those of which the node has direct knowledge) is determined.

The aggregate count score is determined by adding the three scaled, power level node counts and dividing the sum by the scaled total neighbor count. Since this value is merely a percentage of the neighbors directly known by the local node, an adjustment is performed by multiplying the score by the ratio of known neighbors versus the network node count presented by the Sink. This allows the algorithm to favor nodes with global knowledge rather than nodes with a reduced network awareness.

The second variable in Equation 10 is the relative power to Sink score. The relative power score is based on three scenarios, wherein the power levels between the node being scored and the Sink are distinct. The three scenarios are described in Figure 6-9, where the tested node and Sink are equal distances to the local node, the tested node is further than the Sink, or the tested node is closer than the Sink.

![Figure 6-9: Sink Power Level Evaluation Scenarios](image)

The algorithm is designed to favor potential forwarders with the lowest transmit power required to reach the Sink and for whom the local node requires the lowest power to reach. The algorithm, described in Equation 12, begins with a base score of 22% for all nodes. The base score for an...
evaluated node is adjusted based on the required transmit power levels. If the power required to reach the target node is less than or equal to the power required to reach the Sink, from the perspective of the local node, the score is increased by 11%. Conversely, if the power required to reach the Sink is less than the power to reach the target node, then the score is decreased by 11%.

Since a node must evaluate itself, the notion of “Local Power” is introduced into the source code. Local Power is a 0-cost power level, as a node may always reach itself. This allows a node to favor itself when other factors are nearly equal. This does not guarantee that the local node will always pick itself, as the final score is relative to the required Sink power, not other nodes.

Finally, the SINK\text{SCORE} is calculated by multiplying the adjusted score by a constant based on the power required to reach the Sink. This results in a value somewhere between 22% and 99%. The percentage is used as a floating-point value in Equation 10.

\begin{align*}
Base &= 0.22 \quad \text{(A)} \\
Adj &= \begin{cases} 
Base - 0.11 : & \text{Fwdr} > \text{Sink} \\
Base + 0.11 : & \text{Fwdr} \leq \text{Sink}
\end{cases} \quad \text{(B)} \\
\text{SINKSCORE} &= \begin{cases} 
Adj \times 3 : & \text{Power to Sink is LOW} \\
Adj \times 2 : & \text{Power to Sink is MED} \\
Adj \times 1 : & \text{Power to Sink is HI}
\end{cases} 
\end{align*}

Equation 12- Power to Sink Score Calculation

The final value in the dividend is the Percentage of System Power Remaining for the target node. As each OGM contains the percentage of system power for the transmitter, scaled to an 8-bit integer, the local node can divide this value by 255 to recover the floating-point value with a range of [0, 1.0], with discrete steps of 0.39%.

All three dividends added together produce a number whose maximum value is 3.0; therefore, the node scoring algorithm divides the dividend by 3.0 to produce a normalized percentage value. This value is then compared against the score for all other known neighbors. The node identified with the highest score is the Next-Hop neighbor to be advertised by the local node. If there is a

Nelson Henry Powell III
tie, the node with the lowest MAC address is the tie breaker. It should be noted that a node may advocate for itself to be the forwarder.

Once the evaluation is complete, each node creates a RTSEL OGM, with the selected node MAC address populating the Next-Hop field. At this point, if there is are peer nodes that the local node perceives but has not received an OGM directly from the peers, the peers’ addresses are appended to the RTSEL OGM using the NOT_FOUND TLV. Unlike other OGMs, every node generates a single RTSEL OGM.

If there are more than 53 indirect neighbors, the node must wait another round before transmitting a NOT_FOUND TLV with the additional MAC addresses. This problem does resolve itself over time in that not all neighbors are missing the identical set of indirect peers. Therefore, in periods of high levels of collisions, multiple nodes may produce disparate lists of indirect neighbors, forcing a larger set of nodes to rebroadcast their RX POWER OGM in the next round to correct the symmetry issue. Nodes identified by another sensor to retransmit its RX POWER OGM may resolve another node’s indirect neighbor issue.

\[
T_{RTSEL} = [T_{SLOT} \times N \times k_1] \quad \text{(A)}
\]

\[
S_T \sim \mathcal{U} \left[0, \left\lfloor \frac{T_{RTSEL}}{T_{SLOT}} \right\rfloor \right] : S_T \in Z \quad \text{(B)}
\]

Equation 13- Router Selection Time-Period

To transmit their RTSEL OGM, nodes must again fall back on the pseudo-TDMA process. Like the algorithm used in Equation 8, Equation 13 presents the slot selection process for the Router Selection phase. Computing the slot start time in Equation 13, the algorithm returns to Equation 8 (C) and (D). For the purposes of implementation, the \( K \) constant is set to 1.25, and thus the time-period for the Router Selection phase is identical in length to that of the Startup Phase.

Upon receiving a RTSEL OGM from a peer, the receiving node adds the elected node to a list of potential Next-Hop nodes. If the node is already in the list, then the counter for the given node is incremented. The node(s) with the highest number of ballots may become a Next-Hop forwarder. When the Sink transmits a Super Frame, the Number of Forwarders field instructs the member...
nodes of the maximum number of forwarders ($F_N$) allowed in the round. Each node compares the scores of the top $F_N$ nodes to determine the group of potential Next-Hop nodes.

Unlike other phases, the Sink does not “timeout” and start a new phase when the Router Selection phase is complete. The Sink actively receives throughout the Router Selection and Data Exchange phases as a single phase. Only the member nodes view the phases as separate time periods. Therefore, the equation used for calculating the Sink’s timeout is demonstrated in Equation 14.

$$T_{RXPWR} = \left[ T_{SLOT} \times N \times k_1 \right] + \left[ T_{SLOT} \times M \times N \times k_2 \right] + 10ms$$

**Equation 14- Sink Combine Timeout Calculation**

The Sink timeout calculation has three components, the Router Selection timeout, the Data Exchange timeout, and additional overhead time for resynchronization. The first component is identical to the time-period expressed in Equation 13A. The second component is identical to the formula in Equation 15A. The aggregation of the two time-periods ensures the Sink allows both phases to complete. Since the Sink starts its timer before the nodes, it needs to allot additional time for distant nodes to complete their timeout process. For this implementation, 10ms was hardcoded in the waveform.

### 6.4.4 Data Exchange Phase

The *Data Exchange* phase is the last core state in the FSM and notionally begins at the end of the Router Select period. Centralized synchronization of the state change is limited as each node starts its timer based on the reception of the Periodic OGM Super Frame. Nodes are synchronized loosely since all receptions are skewed due to propagation delay. However, the maximum skew is around 300ns, and thus does not severely impact the function of the network.

At the start of the *Data Exchange* phase, nodes examine the group of potential Next-Hop nodes. A node may select a peer or itself as a forwarder for the round. If the node chooses a peer as a forwarder (does not select itself as a forwarder), it must validate the transmit power required to reach the Next-Hop node. If the power required to reach the Next-Hop is greater than or equal to the power required to reach the Sink directly, then the node will transmit its data directly to the
Sink. If the power required to reach the Next-Hop node is less than the power required to reach the Sink, then the node transmits its data to the Next-Hop node.

Data Exchange occurs over a predefined time-period based on Power Collection timeout. Equation 15 expresses the calculations used to define the Data Exchange time-period. As before, the Data Exchange phase relies on a pseudo-TDMA channel access. The number of TDMA slots is based on the number of nodes multiplied by a constant $K$ to provide for extra rounds for the reduction of collisions. For implementation purposes, $K$ is set to 1.25 for 25% more slots. The time-period is the length of a slot multiplied by the number of slots.

However, the number of slots is multiplied by the same multiplier $M$ used in the Power Collection time-out. Nodes do not need to transmit more than one Data Frame; rather the $M$ multiplier allows for enough extra time for the Next-Hop nodes to transmit aggregated data. Unlike the Power Collection phase, the $M$ multiplier defaults to 2 when the number of nodes is less than 53 nodes in the network. Without the minimum value, there may not be enough slots for the forwarders to negotiate the transmission of Forward Frames.

If a node elects itself as a forwarder, the node holds off transmitting data until later in the Data Exchange phase. Nodes back off to collect and aggregate Data Frames from other nodes, and sending the aggregated frame once reducing the total number of transmissions in the round. Non-forwarding nodes attempt to transmit their Data Frames earlier to allow for data aggregation by forwarding nodes.

$$T_{EXCH} = [T_{SLOT} \times M \times N \times k_2] \quad (A)$$

$$S_T \sim \bigcup \left[0, \left\lfloor \frac{T_{EXCH}}{T_{SLOT}} \right\rfloor \right] : \quad S_T \in Z \quad (B)$$

Equation 15- Data Exchange Time-Period

Again, looking at Equation 15, nodes subdivide Equation 15(A) into two sub-phases. The $T_{EXCH}$ time is separated into a 75% time-period and a 25% time-period. The first 75% of the period is allotted for nodes to transmit their data to a Next-Hop address or directly to the Sink. The last 25% of the period is allotted for Next-Hop nodes to transmit aggregated data to the Sink.

Nelson Henry Powell III
All nodes are aware of the preconfigured time-period ratio and determine the time-period based on the number of nodes in the network, as advertised by the Sink. Therefore, all non-forwarding nodes can schedule a sleep period before and after their transmission. This allows BATSEN to replicate the power savings found in the LEACH protocol without the need for tightly controlled TDMA slotting. Only forwarding nodes and the Sink need keep on their RF hardware.

At the end of the Data Exchange phase, the Sink will transmit either a RX Power Super Frame or Periodic OGM Super Frame. If the Sink received a RTSEL OGM with a NOT_FOUND TLV or received a NULL OGM from a node not accounted for, the Sink will transmit an RX Power Super Frame, supporting the LNE operations. If the Sink does not detect the need for LNE operations, or all nodes have successfully exchanged link-power information, the Sink will have the network skip the Power Collection phase and immediately restart the Router Selection phase. The Sink does this by transmitting a Periodic-OGM Super Frame.

As every transmission by a node contains the percentage of system power remaining in the node, peers can update the power percentage item per distant node based on the values in Data Frames and RTSEL frames. It is possible that a node’s notion of a peer’s power remaining is out dated by the time the local node begins to evaluate the potential Next-Forwarders. If the Sink does not force all nodes to follow the transition 14 path, the percent power remaining used for Router Selection will always be 1-round behind. The effects of the latency should be negligible as the algorithm allows for rotation of router responsibilities.

6.4.5 Late Net Entry Operations
In a real-world deployment, it is unlikely that all nodes are powered on at the same time. It is also possible that an administrator enables the Sink prior to some of the nodes. In either case, there is a need for LNE operations: a method for late nodes to enter the network and function as a member node. The NULL OGM and the NOT_FOUND TLV in the RTSEL OGM provides the LNE mechanism.
When a node comes online late, it will have missed the NULL Super Frame. Receiving any OGM will force the node into the LNE process. The node follows transition 2 and transmits its NULL OGM for the Sink and peer nodes to collect. The node then listens for any Super Frames and OGMs to determine the network’s FSM state. Based on the Super Frames, the LNE node may follow transitions 3, 4, or 5 to the appropriate state in the state machine.

Nodes that receive the late node’s NULL OGM add the node’s MAC address to a list of indirect neighbors. When creating the RTSEL OGM, the late node’s MAC address is added to the NOT_FOUND list. This signals the Sink to follow transition 14 rather than transition 13, repeating the Power Collection process such that the new node can build up its internal peer tables, and existing nodes can begin to evaluate the score of the new node. This process may occur at any point within the FSM, and thus synchronizes and merges new nodes into an existing BATSEN network.

6.5 Protocol Timing Analysis
Figure 6-10 illustrates the timeline for BATSEN in a similar manner as Figure 5-7 does for LEACH.

The length of a BATSEN round can be calculated by laying out the FSM states into a timeline and adding the time periods as defined from the perspective of the Sink. Time-periods are defined as a function of the number of nodes (N), as there is no notion of a CH in BATSEN. Time constants or constant multipliers are literally defined within each phase’s time-period definition. Slot lengths are defined as CSMA/CA based, where T_{CSMA} is 7ms. The T_{CSMA} slot occupies approximately 5.2ms for transmitting a maximum frame (127 Bytes) and another 1.8ms for the carrier sense function. For 100 nodes, the length of the BATSEN epoch is estimated as 5.995 seconds for the first round and 5.113 seconds for subsequent rounds. If nodes do not transmit the NOT_FOUND TLV, and the Sink is allowed to skip a Power Collection phase, then the round length is estimated at 2.985 seconds.
Figure 6-10: BATSEN Protocol Timeline
7 Results
The primary objective of this thesis is to conserve power required for channel access in order to support power hungry security primitives. Therefore, when comparing LEACH and BATSEN, the first order of business is the evaluation of network lifetime.

Figure 7-1 presents the Network Lifetime for both protocols against the distribution method and Sink location. Each X-gridline is provided a 3-character title representing the Protocol, the Distribution Methodology, and the Sink Location. The Protocol field is of the set \([B,L]\), where \(B\) represents the BATSEN protocol and \(L\) represents the LEACH protocol. The Distribution field is of the set \([R,U]\), where \(R\) represents pure random distribution and \(U\) represents the uniform distribution model. Similarly, the Sink Location field uses a code in \([B,C]\), where \(B\) represents the bottom of the unit circle and \(C\) represents the center of the circle. As an example, the BRB x-gridline represents the scenario testing the BATSEN protocol, with Random distribution, and the Sink located at the bottom of the circular test space.

![Network Lifetime for All Scenarios](image)

**Figure 7-1: BATSEN vs LEACH, Network Lifetime for all Scenarios**

The BATSEN protocol lasts about 940 seconds on average, regardless of scenario, while the LEACH protocol lasts about 260 seconds, or 27% of the BATSEN lifetime. To understand the disparity, the first obvious disparity lies in the length of a round. LEACH was calculated to have
a round length of 2.7127 seconds for a 5 CH network, whereas BATSEN is estimated to use a 5.113 second round, with the possibility of shrinking to a 2.985 second optimized round.

Figure 7-2 provides the empirical results of the average round length for each protocol. The LEACH protocol holds true to the estimated time per round, as the Secondary Advertisement phase is used frequently to compensate for the Threshold algorithm. The BATSEN protocol varies in round length per scenario, thus the curve in Figure 7-2 averages the length of a BATSEN round. It is noted that the average round length holds around 3 seconds per round. Therefore, the LEACH round length is about 90% of the BATSEN round length. This does not translate directly to the lifetime extension of 360%.

Looking at the components of the LEACH protocol, given 100 nodes and 5 CHs per round, the two Advertisement Phase and the Schedule Phase account for 210ms and 1.7955sec respectively. The remaining time 707.2ms is dedicated to the Data Phase and Sink Data Phases. Of the total 2.7127seconds per round, only 26% of the protocol allows nodes to sleep. The results of attempting to create a decentralized network forced additional synchronization and overhead to the point of diminishing returns.

BATSEN on the other hand, has a maximum round length of 5.995 seconds for the first round. Rounds drop to 5.113 seconds after the first round. The extra RX Power process increases the
round length 2.128 seconds, but the extra length ensures a rapid convergence of the network population. After a number of rounds establishing bi-directional relationships between peer nodes, the protocol drops to only 2.985 seconds per round.

The Data Exchange phase lasts for 2.1 seconds of the round. In the early rounds of the protocol, BATSEN allows nodes to sleep for upwards of 40% of each round. After establishing the bi-directional relationships, and the round length drops, the same Data Exchange phase occupies over 70% of the round length. It is this feature of the protocol that allows the BATSEN to outlive the LEACH protocol.

Unfortunately, BATSEN also suffers performance reduction due to topological changes just like the LEACH protocol. LEACH tends to lose about 16% of its lifetime when network layouts follow a Uniform distribution rather than a true Random distribution. BATSEN does not experience a loss in overall lifetime of the network, as it is stable across all four scenarios; but it does see earlier First Node Death times. Depending on the scenario, First Node Death occurs 10 to 15% earlier using a Uniform distribution of nodes.

Notable is the extreme difference between the First Node Death times based on Sink location in the simulated space. Unlike LEACH, BATSEN is impacted by the physical location of the Sink with respect to the member nodes. The first node in BATSEN dies 50% earlier when the Sink is placed on the circumference of the unit circle. This implies that other nodes will die early as well. Fortunately, the data shows that the 50% Network Dead point is stable regardless of Sink location, holding at approximately 1% before the entire network dies. This indicates that most nodes run out of power nearly simultaneously.

Figure 7-3 and Figure 7-4 present an alternate perspective of Figure 7-1. These bar graphs present the 100%, 50%, and First Node Dead points for the protocols along with the standard deviation illustrated as error bars. Both protocols, regardless of Sink position and network layout are stable, save only for the First Node Dead time for the BATSEN protocol. All error bars (except for the FND time) are less than 1% for the given average value. This indicates highly stable and reliable...
protocol implementation, but also guarantees suboptimal performance for LEACH regardless of scenario.

![Network Lifetime Sink in Center](image)

**Figure 7-3: Network Lifetime BATSEN vs LEACH, Sink at Center of Unit Circle**

![Network Lifetime Sink at Bottom](image)

**Figure 7-4: Network Lifetime BATSEN vs LEACH, Sink at Bottom of Unit Circle**

For BATSEN, the standard deviation varies up to 30% of the bar for both the Sink in the Center and Sink at the Bottom scenarios. The large error indicates the BATSEN protocol’s sensitivity to the layout, as error is introduced by randomly assigning positions. The disparity between the average First Node Dead times points to the BATSEN protocol’s sensitivity to Sink position.
BATSEN does not support the CH concept. Rather, node’s vote on desirable Routers, or Forwarders as the protocol operates at Layer 2 of the OSI model. For testing purposes, the BATSEN MAC used the SensorHelper CH callback method to track the number of elected Forwarders for a given round. Figure 7-5 and Figure 7-6 present the resulting CHs per Rounds curves for both protocols based on Sink location.

BATSEN attempts to restrict the number of forwarders to 5% of the nodes in the network. Since election is a cooperative and decentralized effort, there is no guarantee that any two nodes derive the same Next-Best Hop list. This is evident in that BATSEN produces upwards of 12 Forwarding Nodes per round when the Sink is at the center of the circle, and around 7 Forwarding Nodes when the Sink is on the circumference.

Looking closer at Figure 7-5, BATSEN experiences a dip in the average number of Forwarders between round 100 and 150. This is due to nodes re-electing the same peer nodes excessively to the point of severe power reduction. The nodes commonly elected as forwarders do not fail at this point, rather other nodes detect the reduced power capacity and the election criteria forces all nodes to elect new forwarders.

At approximately round 200, all nodes are experiencing heavy power drainage. Only a few nodes, typically further away from the Sink, have the capacity to function as a Forwarder. Nodes further
away from the Sink typically have the power remaining as they are less likely to be elected from the start of the network due to the power level required to reach the Sink, as the algorithm favors closer nodes. It is around the same time that the average First Node Death event occurs.

Figure 7-6 presents the graph of CH and Forwarder counts per round for the Sink located on the circumference of the unit circle. In this case, BATSEN achieves a lower Forwarder count, on average, closer to the desired 5%. The reduction in Forwarders is explained by the election algorithm favoring nodes closer to the Sink. By shifting the Sink to the periphery, nodes located in the same half of the circle are more likely to achieve Forwarder status. But with a concentrated population, far fewer nodes need to function as Forwarders per round.

Similar to the curves in Figure 7-5, BATSEN experiences a dip starting at round 125. Unlike the Sink positioned in the center scenario, this dip is caused by the First Node Death event. As elected nodes are concentrated to one side of the test area, fewer nodes are allowed to share the burden of acting as a forwarder. This causes faster energy expenditure than the Sink in Center scenario. Once the closer nodes die, nodes in the medium and high-power ranges become more likely to be elected as forwarders, thus spreading the energy expenditure throughout the network and maintaining the long network life.
Figure 7-7 and Figure 7-8 support the findings of Figure 7-5 and Figure 7-6, illustrating the average number of DC nodes per round. LEACH allows nodes to resort to DC operation when they are unable to join a cluster during the Schedule phase. BATSEN is much more liberal in that nodes vote on their most desirable Forwarder; but after the election process is complete, if their own power level required to reach the Sink is less than or equal to the power level required to reach the Forwarder, they switch to a DC operation.

Starting with Figure 7-7, the LEACH protocol experiences approximately 4% DC nodes, except during the end of a cycle. Recalling the cyclic drop in active CHs (Figure 5-13 through Figure 5-16), the LEACH protocol fails to consistently select the required number of CHs at the end of each cycle. If there are not enough CHs to support the network, then nodes are forced to resort to the DC mode to get their information to the Sink. This is seen by the periodic spike in DC nodes for the LEACH curves.

BATSEN, being more liberal with the use of DC nodes, experiences approximately 20% to 35% of the nodes acting in the DC mode of operation. The Random distribution experiences a higher level of DC activity in that nodes are more likely to be clustered physically. With the Sink at the center of the unit circle, clumps of nodes located close to the Sink do not need to rely on a Forwarder. With uniform distribution, the clustering of nodes is reduced, thus more nodes exist further away from the Sink and must rely on Forwarders to reduce their power expenditure.

![DC Nodes per Round with Sink at Center of Circle](image)

Figure 7-7: BATSEN vs LEACH, Direct Connect Nodes per Round, Sink at Center
With respect to Figure 7-5, Figure 7-7 shows the rapid increase in DC nodes around round 200. This correlates to the rapid drop in Forwarders around round 200. As the network shifts to nodes further away from the Sink to operate as Forwarders, nodes closer to the Sink will save power by selecting to send their data directly to the Sink. Only other nodes of a similar distance to the Sink would save power by sending their data to a forwarder.

Figure 7-8 presents similar information with respect to Figure 7-6. Again, the DC node count for LEACH is cyclic and experiences approximately the same level of DCs per round as the Sink at the center scenario. BATSEN, however, experiences a 50% reduction in DC nodes with respect to the Sink at the center scenario. With the Sink on the periphery of the test area, only 10% to 13% of the nodes typically operate in the DC mode.

Another difference between the Sink location scenarios is the fact that BATSEN DC nodes increase when the network is deployed using the Uniform distribution model. With the Sink located to one side of the test area, and Random distribution allows for physical clustering of nodes, distant nodes are more likely to have more nodes to choose from to elect as forwarders. It is not a significant difference, but it is noticeable.

Figure 7-8 also compliments Figure 7-6 in that the FND event is evident through the rapid increase in DC node near round 125 for Uniform distribution and round 140 for the Random distribution. At these moments, the first node to die was used excessively as a Forwarder. The nodes that relied on the first node to die switch over to DC operations until a new node’s score is significantly high enough to draw DC nodes back to relaying operations.

Just as in Figure 7-7, when the closest nodes die, medium range nodes tend to switch to DC operations, if not elected Forwarders, as seen by the spike in DC nodes around round 250. Only maximum distance nodes elect Forwarders, as was evident in the drop in Forwarders in Figure 7-6. Power savings is minimized at this point, but still allows for the network to function for the optimal 940 rounds.
Figure 7-8: BATSEN vs LEACH, Direct Connect Nodes per Round, Sink at Bottom

Figure 7-9 illustrates the actual data throughput of the networks for both protocols across the four scenarios. LEACH is consistent at the 37 packets/second. BATSEN varies over time based on its reliance on CSMA/CA to deliver data to the Sink. LEACH uses TDMA which occurs at the latter half of the round’s cycle. Regardless of the location of the TDMA period within the protocol, the data rate is guaranteed, and only fluctuates when nodes resort to DC operations.

Figure 7-9: BATSEN vs LEACH, Packet Rate per Round

BATSEN uses CSMA/CA for all communications. To improve the probability of reception, nodes attempt to randomly select hybrid-TDMA slots rather than selecting a pure random time within a
phase’s time-period. However, there is still significant loss. The graph shows BATSEN providing throughput ranging from 23 to 26 packets/second. When the network requires RX Power frame exchanges, a round takes 5.1 seconds. For 100 nodes, the maximum throughput is 19.5 packets/second. This packet rate is seen early on in the network, but rapidly increases after the network stabilizes.

Once bi-directional relationships are established, the round is reduced to 2.985 seconds. For 100 nodes, the maximum data rate is calculated as 33.5 packets/second. At 25 packet/second, BATSEN consistently suffers a 25% loss in data. Worse yet, the position of the Sink impacts the throughput of the network. When the FND event occurs, there is a drop-in throughput to approximately 17 packets/second, which only slightly recovers after the network selects new Forwarders. This may be caused by the incomplete implementation of the neighbor presence vector. Without detecting the loss of a neighbor, some nodes will continue to incorrectly elect a dead neighbor as a forwarder.
8 Conclusion

8.1 Key Contributions of this Work
This work provides both practical benefits for future experimental protocol testing and broadens the area of research methodologies in wireless protocol design.

The purpose of this research was to fuse technology from one field of wireless networking into another, in an attempt to discover new power savings techniques. In the process of conducting the research, a generic framework for simulating wireless sensor network protocols was developed. This framework provides an API for future protocol implementations and the necessary interconnections with other adjacent layers to support high-fidelity simulations.

During the research phase of the project, it was discovered that much of the contemporary research relied on simulation techniques focused on one level of the OSI model. Researchers either designed to a single level or relied on test environments incapable of incorporating high-fidelity simulations of the adjoining OSI layers. Theoretical models may provide optimal results for the module under test but fail to perform under system level test environments.

This paper broadens the area of research methodologies for waveform design by following a more holistic approach. Rather than immediately developing mathematical models, this project identified the design assumptions of contemporary works at each level of the OSI model. The validity of the assumptions was checked and scrutinized. The results of the research were used to drive design decisions in conjunction with mathematical modeling. Using this systemic approach to waveform development ensured that each layer operated sympathetically to produce an optimal system rather than an optimal algorithm.

8.2 Summary of Findings
This research sought to develop a new wireless sensor network routing protocol, based on the BATMAN protocol, in an attempt to improve power consumption over the traditionally accepted theoretical protocols. Power optimization of channel access and data exchanges in wireless sensor networks is paramount. Power is a premium in wireless sensors, particularly in the age of constant evolving cyber-attacks. Sensors must perform their primary function, sensing and reporting their
environment; but now must provide Information Assurance features to protect their data and the stability of the network.

BATSEN and LEACH, combined with an 802.15.4 PHY, have nearly identical round lengths, BATSEN being only 10% longer than LEACH. It is clear from the experimentation, that a realistic implementation of LEACH suffers from a longer setup-phase causing unnecessary power loss. LEACH requires the extended setup-phase operations to support network synchronization, guarantee cluster membership, stabilize CH selections, and mitigate intra-cluster channel access issues. Moreover, the additional overhead required at the MAC layer to support a functional DSSS PHY goes unchecked without the proper support at the PHY layer.

This thesis has demonstrated that one cannot relegate the mathematical advantage of consolidating traffic in an effort to reduce power consumption. This thesis has also shown that a holistic approach to system design invalidates protocols that do not account for system level interactions. This is evident in the number of variations on the LEACH algorithm, each asserting their own assumptions to make up for one or more failures in the base algorithm.

BATSEN, using CSMA/CA for all communications, provided a 197% longer steady-state operation over LEACH while reducing the setup-phase operations by 56%, allowing for dramatically increased network lifetime over LEACH. The development of BATSEN used a more holistic and empirical approach to the design, using the mathematical models as a goal where practical. This resulted in superior power performance at the cost of data throughput.

BATSEN is based on decentralized layer 3 routing combined with a node-based voting mechanism to perform a simplistic swarming algorithm for Forwarder selection. The OGM message processing and FSM allow for Late Net Entries, which alleviates many of the a priori requirements of traditional sensor network protocols. Moreover, BATSEN is not tied to dedicated data transmission paths. BATSEN allows nodes to determine the Next Best Hop, even if the next hop is a direct path to the Sink, allowing for more efficient use of minimum transmit energy techniques.
BATSEN is not without its issues, however. Though it does not noticeably affect overall network lifetime, the Sink’s position within the network topology greatly impacts the First Node Death event timing. This is a side effect of not incorporating the node presence or history bit-vector into the Best Next Hop scoring algorithm. This was seen in debug as many nodes continued to select a dead node as the Best Next Hop.

Besides detection of dead nodes, the Best Net Hop scoring algorithm should account for frequency of Forwarder selection. Again, when the Sink is positioned on the periphery of the network, nodes closer to the Sink tend to die faster. Detecting high frequency Forwarders would allow nodes to reduce their heavy reliance on a subset of nodes. This should spread the power dissipation better than the current scoring system and help avoid killing nodes in close proximity to the Sink too soon.

BATSEN also suffers the reduced data throughput rates traditionally experienced with CSMA/CA based channel access protocols. LEACH provides a guaranteed data rate based on the length of the round and the number of nodes in the network through the use of TDMA. BATSEN may be able to leverage the use of spread-spectrum technologies or fine grained transmit power level control to reduce the probability of collision, thus increasing the packet throughput rate.

Overall, BATSEN provides a new path of exploration in the realm of Wireless Sensor Networks. Having a flexible, efficient, and realistic waveform based on sound design principles presents the industry with new opportunities in secure wireless communications. Providing more residual power and network management opportunity to a designer will allow for testing various methodologies to secure channel access and data transfers for future WSNs.

8.3 Future Work
There are many tests and protocol modifications that may be implemented to improve the analysis of BATSEN against LEACH, as well as other Sensor Network Protocols.

Having a pre-built 802.15.4 PHY allowed for rapid development of both the LEACH and BATSEN protocols, but some fidelity loss was traded to test base functionality. For future
development and testing, the lr-wpan model should incorporate an accurate DSSS model to perform and detect real signal levels during TDMA slots. The current simulation relied on separate channels with no co-site interference, which relegates the effects of multiple transmitters.

The energy dissipation model may also be too generic. It would be interesting to acquire a real sensor and measure the power dissipation and determine the transmit power level precision. All future research would benefit from updating the SensorMac base class’ energy models and allow test results to prove more realistic results based on actual hardware.

The lr-wpan PHY implementation supports both Energy Detection (ED) and Clear Channel Assessment (CCA) for the Collision Avoidance operations. The current state of the SensorMac only provides the CCA functionality, which forces a node to look for signals used by the MAC. If interference is to be modeled, non-MAC based signals must be included, but would not be detected by the CCA module. Therefore, MACs would need to include support for both ED and CCA operations during the CSMA/CA based states.

In terms of the current test scenarios, additional test parameters may be needed to fully illuminate the differences between BATSEN and LEACH. One might query the mobility information attached to each node to determine the maximum, average, and standard deviation of node-to-forwarder distances. This would confirm the theory that in-network processing would allow protocols to disregard co-located sensor data.

Distance testing should also be coupled with the creation of a Sensor Application. NS-3 provides standard APIs for higher layers to send and receive data to and from lower layers. The SensorMac class conforms to these APIs to allow for future expansion of the sensor test suite. Studying the behavior and modeling a real sensor would allow for real data transference rather than tracked sequence number frames. This would allow for a true test of the in-network processing proposed by Heinzelman [7].

The LEACH protocol developed for this thesis could use improvements in various portions of the waveform. The current approach to CH assignment added an additional state in the FSM, thus
extending the round length and reducing the protocol’s performance. A better algorithm for CH self-election needs to be developed to ensure that the desired CH count is achieved every round. Moreover, a method is needed to ensure that CHs are not co-located.

The LEACH cluster membership process allows for too many Direct Connection nodes. If a method could be developed to ensure equal distribution of members across all clusters, the power dissipation should likewise be distributed more evenly. It should be noted that reducing collisions during the cluster membership establishment may require an increased Setup Phase, thus reducing the power efficiency of the network. In all, this enhancement needs to be evaluated based on protocol deployment requirements versus power efficiency. It is by no means a simple modification.

BATSEN likewise has many improvements that could be implemented for future research efforts. For starters, the scoring algorithm could be improved by accounting for the frequency of peers getting elected as Forwarders and comparing SINR of Forwarders to ensure elected nodes do their best to avoid co-location.

Though the BATSEN protocol implemented a bit-vector tracking system for neighbor presence, the presence vector was not incorporated in the scoring algorithm which allowed nodes to continually pick dead nodes, as their power levels are never updated post mortem. The presence bit-vector needs to be connected to the scoring algorithm to determine if the network lifetime is extended and the level of Direct Connection nodes is reduced toward the end of the network lifetime.

BATSEN relied on a number of constants in its various state timeout algorithms. These constants should be varied and tested for optimal performance. It is unclear as to how much the different timeouts affect the network. Curves should be run against the protocol to determine which constant should change, and under what conditions the changes provide enhancement or degradation of network performance.
Beyond the testing of protocols as they are, BATSEN should be expanded to fully support Late Net Entry and Multi-Hop test scenarios. There are other variants of LEACH that can be implemented to compare against the features inherent to BATSEN. The LEACH protocol, as implemented, does not support either feature, thus other protocols or LEACH variations must be implemented.

Besides comparisons to other protocols, BATSEN needs security mechanisms. The base protocol provides for an optimized power utilization and the coveted Late Net Entry operation; but it does not include any primitives for security in the network. Wireless Sensor Networks need protective measures similar to Enterprise networks: Authentication, Authorization, Confidentiality, Integrity, and Availability, and Non-Repudiation.

Future work should include modifications to the existing BATSEN frame formats and messaging procedures to incorporate various protection technologies. Technologies include, but are not limited to: encryption, secure hashing, digital certificates, key rotation, and pseudo AAA services.
9 Bibliography


Nelson Henry Powell III


