3-10-2010

Analysis of a Rumor Routing Protocol with Limited Packet Lifetimes

Peter R. Francik

Recommended Citation

Francik, Peter R., "Analysis of a Rumor Routing Protocol with Limited Packet Lifetimes" (2010). *Theses and Dissertations*. 2010.
https://scholar.afit.edu/etd/2010

ANALYSIS OF A RUMOR ROUTING PROTOCOL WITH LIMITED PACKET LIFETIMES

THESIS

Peter R. Francik
Captain, USAF

AFIT/GE/ENG/10-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/10-09

ANALYSIS OF A RUMOR ROUTING PROTOCOL WITH LIMITED PACKET LIFETIMES

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Peter R. Francik, BS
Captain, USAF

March 2010

AFIT/GE/ENG/10-09

ANALYSIS OF A RUMOR ROUTING PROTOCOL WITH LIMITED PACKET LIFETIMES

Peter R. Francik, BS
Captain, USAF

Approved:

Rusty O. Bald
Dr. Rusty O. Baldwin (Chairman)

3/16/10
Date

Dr. Brett J. Borghetti (Member)

3/16/10
Date

Barry Mullins
Dr. Barry E. Mullins (Member)

16 Mar 10
Date

AFIT/GE/ENG/10-09

Abstract

Wireless sensor networks require specialized protocols that conserve power and minimize network traffic. Therefore, it is vitally important to analyze how the parameters of a protocol affect these metrics. In doing so, a more efficient protocol can be developed.

This research evaluates how the number of nodes in a network, time between generated agents, lifetime of agents, number of agent transmissions, time between generated queries, lifetime of queries, and node transmission time affect a modified rumor routing protocol for a large-scale, wireless sensor network. Furthermore, it analyzes how the probability distribution of certain protocol parameters affects the network performance.

The time between generated queries had the greatest effect upon a network's energy consumption, accounting for 73.64% of the total variation. An exponential query interarrival distribution with a rate of 0.4 queries/second/node used 25.78% less power than an exponential distribution with a rate of 0.6 queries/second/node. The node transmission time was liable for 73.99% of the total variation in proportion of query failures. Of three distributions, each with a mean of 0.5 seconds, the proportion of query failures using a Rayleigh transmission time distribution was 14.23% less than an exponential distribution and 18.46% less than a uniform distribution. Lastly, 54.85% of the total variation in the mean proportion of time a node is uninformed was a result of the time between generated agents. The mean proportion of time a node is uninformed using an exponential agent interarrival distribution with a rate of 0.005 was 6.59% higher than an exponential distribution with a rate of 0.01.

Acknowledgments

I am profoundly indebted to my research advisor, Dr. Rusty Baldwin. This thesis would not have been possible without his steadfast guidance and support. I am also grateful to Dr. Jeffrey Kharoufeh for his patience and input. Lastly, I offer my regards to my committee members, Dr. Brett Borghetti and Dr. Barry Mullins, for their selfless contributions to improve the quality of this research. Thank you!

Peter R. Francik
March 2010

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

ANALYSIS OF A RUMOR ROUTING PROTOCOL WITH LIMITED PACKET LIFETIMES

1. Introduction

1.1. Introduction to Wireless Sensor Networks

The demand for real-time data has exploded as technological advancements produce devices that are physically smaller, faster, and cheaper. Among such devices are autonomous sensors that provide data in a simple and cost-effective manner. As the uses for these sensors grow, so does the need for them to communicate with each other in ever-increasing numbers. That, coupled with applications requiring mobile sensors, led to the development of wireless sensor networks (WSN). Today, WSNs are embedded in structures, machinery and environments, aiding in such tasks as averting disastrous structural failures, conserving natural resources, providing improved emergency response, and enhanced homeland security [L04].

WSNs contain homogeneous nodes that can self-organize into an ad hoc, multi-hop wireless network. The nodes, an example of which is shown in Figure 1, typically consist of at least one sensor, an on-board processor, memory, short-range radio, and a battery. After deployment, it is unlikely a node's battery will be recharged, thus power consumption is a primary concern for any WSN.

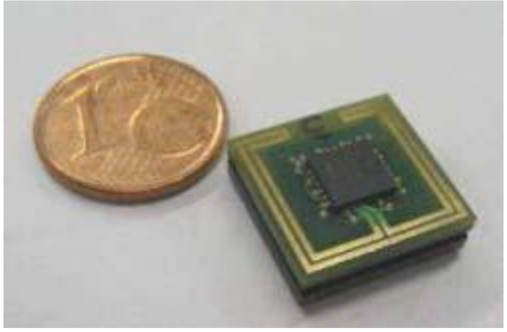

Figure 1: Example of a wireless sensor node [EETA07]

1.2. Problem Statement

Protocols for WSNs are designed to conserve energy. A modified rumor routing protocol, [MBK+08], did so by limiting the lifetime of packets traversing the network. The parameters influencing the performance of the network, however, were not fully evaluated. Furthermore, the protocol assumed exponential distributions for each packet-related parameter and did not examine the effects of other probability distributions.

1.3. Research Goals

This research determines the effect various parameters have on the protocol. Specifically, this research:

1. Updates the modified rumor routing simulation so its packet-related parameters can be modeled by an arbitrary distribution.
2. Uses OPNET, a discrete-time network simulator, to analyze the effect each parameter has upon the performance of a WSN, focusing specifically on the mean rate of packet arrivals per node, proportion of query failures throughout the network, and the mean proportion of time each node is uninformed.

1.4. Thesis Overview

This chapter introduces WSNs and discusses the constraints that guide their design. The need to evaluate the effect of each factor of a modified rumor routing protocol is discussed, and the research goals outlined. Chapter 2 provides a review of relevant literature. Chapter 3 defines the methodology and identifies the system under test. It also defines the performance metrics being measured and identifies the key factors that affect the system’s performance. In Chapter 4, the model developed in OPNET is described, its performance is verified against the original protocol [MBK+08], and the effects each factor has upon the performance of the system is analyzed. Chapter 5 summarizes the results and discusses the contributions of this research.

2. Background

Wireless sensor networks consist of a large number of densely distributed nodes that self-organize into a multi-hop wireless network. Nodes are typically homogeneous and consist of at least one sensor, an on-board processor and memory, short-range radio and are battery powered. WSNs gather information for a variety of military and civilian applications such as monitoring natural ecosystems, battlefields, and man-made structures.

WSN nodes, although designed to have a long operational lifetime, are likely isolated after deployment and thus have limited resources such as memory, processing speed, and power. These limits restrict a node's transmission range and data rate, leaving them prone to failure. With each failure, the WSN's connectivity and effectiveness decreases, shortening the lifespan of the WSN. Therefore, WSNs require protocols that differ from traditional wireless networks.

Ideally, a WSN must be able to configure itself without prior knowledge of the network topology. It must be scalable and adapt to node additions and failures. It must provide guaranteed delivery of data and fair channel access to all nodes. Finally, it must minimize individual node energy consumption to prolong the network's life. In reality, however, it is difficult to attain all of these requirements due to a node's scarce resources. Research into new medium access control (MAC) protocols and routing algorithms, however, have made great improvements in this area.

2.1. MAC Protocols

A MAC is important to the successful operation of any network. It is responsible for regulating how a medium is shared and ensuring no two nodes interfere with each other's transmissions and cause packet collisions. This is especially important for WSNs because every re-transmission wastes energy. One of the most well-known wireless MAC protocols, IEEE 802.11 [LAN97], is widely used in ad hoc wireless networks due to its simplicity. Unfortunately, 802.11 was designed to maximize throughput, minimize latency and provide fairness, giving little regard to energy consumption. As a result, an 802.11 node's radio is always transmitting, receiving or listening to its neighbor's transmissions. A node that is actively listening while no packets are being sent to it wastes up to half as much energy as when transmitting [VL03]. This becomes more apparent as node density and network traffic increase.

Another factor WSN MAC protocols consider is scalability. Nodes will fail over time, new nodes may be added, or environmental changes may temporarily prevent communication between nodes. The MAC must adapt to these changes. Additional attributes to consider, although not as important, are fairness, latency and throughput. Considering these factors, several MAC protocols have been developed that are either contention-based, schedule-based, or a hybrid of the two.

2.1.1. Contention-Based MAC Protocols

Contention-based protocols use variations of carrier sense multiple access (CSMA) techniques. The fundamental characteristic of CSMA is a node listens to the network's shared transmission medium before attempting to transmit. If it detects a transmission in progress, it will wait until that transmission is complete before trying again. Contention-based protocols, such as S-MAC and T-MAC, minimize four sources of energy consumption. The first is *idle listening* in which nodes are kept awake to actively listen for traffic that is not present. Similarly, *overhearing* occurs when an idly listening node picks up broadcasted packets not addressed to it. *Collisions* force a node to retransmit its data, consuming at least twice the energy for the same data. Finally, *protocol overhead* wastes energy and resources by transmitting and receiving large control packets.

2.1.1.1. S-MAC

Sensor-MAC (S-MAC) [YHE02], one of the first protocols designed specifically for WSNs, uses three techniques to minimize energy consumption. The first, *periodic listen and sleep*, has nodes periodically enter a sleep mode, where they turn off their radio and set a timer to wake themselves. Once awake, a node listens for other nodes attempting to communicate before returning to sleep. In this manner, S-MAC reduces idle listening as well as overhearing. Nodes initially listen for their neighbors' schedules. If none are received, a node randomly chooses a sleep schedule and broadcasts it to its neighbors. If a schedule is received, and the node has not already created its own schedule, it adopts that neighbor's schedule. If a node receives a schedule, and it has already created its own schedule, it will consolidate them into a single schedule. In this manner, *virtual clusters* of nodes are formed between neighbors with the same schedule, allowing efficient broadcasts and negating the need to maintain a schedule for each individual neighbor [VL03]. Furthermore, schedules are periodically synchronized among neighbors to prevent long-term clock drift, as well as to adjust for changes in the

WSN. Transmissions take precedence over a node's sleep schedule and a node will not sleep until a transmission is complete.

The second technique addresses *collision and overhearing avoidance*. S-MAC adopts a contention-based scheme similar to IEEE 802.11, including both virtual and physical carrier sense and request to send (RTS)/clear to send (CTS) exchange, to avoid collisions. Virtual carrier sensing includes a duration field in each transmitted packet, indicating the time remaining until the transmission is complete. Thus, a receiving node knows how long to remain silent before transmitting. Additionally, each node performs physical carrier sensing by listening to the medium for transmissions. If both the virtual and physical carrier sense indicates no transmissions, the node is free to transmit. Overhearing is minimized by nodes sleeping upon hearing a RTS or CTS packet between other nodes. In this manner, neighboring nodes only receive the small RTS/CTS control packets and avoid the much longer data packets.

The final technique S-MAC employs is *message passing*, which efficiently transmits long messages. If a long message is sent as a single packet, it risks becoming corrupt, thus requiring the packet to be retransmitted. On the other hand, fragmenting the message creates large control overhead, resulting in a longer delay. S-MAC fragments long messages into smaller fragments, and transmits them in a burst. In this manner, the medium is reserved for all the fragments using only one RTS and CTS packet. With each fragment transmission, the sending node waits for an acknowledgment (ACK) from the receiving node. If it does not receive an ACK, it will retransmit the fragment and extend the reserved transmission time in the duration field to account for the retransmission. Using overhearing avoidance, a neighboring node will sleep upon hearing a RTS or CTS packet until all the fragments have been transmitted, thus reducing switching control overhead. A node that wakes up while fragments are being transmitted will know how long to return to sleep based upon the duration field of the fragment.

Although S-MAC successfully reduces a node’s energy consumption, it does so at the cost of throughput and latency. A node is unable to transmit while asleep, thus throughput is reduced. Further, an event could occur while a node is asleep, but be queued until the node awakens, resulting in an increased delay. Additionally, as the network size increases, nodes must maintain more schedules and incur additional overhead, thus resulting in a shorter lifespan. Finally, S-MAC ignores fairness by allowing nodes with more data to send to monopolize the medium while nodes with shorter packets wait for the medium to be free.

2.1.1.2. T-MAC

The Timeout-MAC (T-MAC) [VL03] protocol improves S-MAC in the area of idle listening. It assumes latency requirements and buffer space are generally fixed, but that message rates vary. Under these assumptions, S-MAC’s periodic listen and sleep cycle is no longer optimized. To adjust for a variable message rate, T-MAC nodes transmit messages in bursts of variable length.

T-MAC initializes similarly to S-MAC until each node has a sleep schedule. Nodes periodically wake up to communicate with their neighbors and stay awake until *activation events* cease for a period of time. These events include the firing of a periodic frame timer, the reception of data, the sensing of communication on the radio, the end-of-transmission of a node's own data packet or acknowledgement, or knowing that a data exchange of a neighbor has ended [VL03].

T-MAC avoids collisions using a contention-based scheme, but does not use the traditional method of increasing the contention interval. Because every node transmits its queued messages in a burst upon awakening, the medium becomes saturated and the traffic load remains relatively high. Therefore, a transmitting node's RTS begins by listening for a random time with a fixed contention interval, even if a collision has not occurred. If the node fails to receive a CTS in reply, the node resends the RTS. If it again fails to receive a CTS, the transmitting node quits and goes to sleep. T-MAC does not use overhearing avoidance when maximum throughput is required. If a node sleeps upon hearing a RTS or CTS packet, it may not hear other control packets, thus reducing maximum throughput.

A side-effect of T-MAC is its susceptibility to the *early sleeping* problem when traffic travels in a unidirectional path. This problem is manifest when a node is unable to transmit to neighbor A due to overhearing neighbor B send a CTS to a different node. While the node waits to transmit, it is possible neighbor A will go to sleep, at which point the node will have to wait until the next contention cycle to transmit. There are two solutions to this problem. The first involves the node sending a *future request to send* (FRTS) packet upon being trumped by neighbor B. In this manner, the neighbor B waits an extra amount of time to avoid its message being corrupted by the FRTS packet. At the same time, the neighbor A receives the FRTS packet and knows not to go to sleep. The second method allows a node that has been trumped to re-trump the original node. If a node's buffer is nearly full and it receives a RTS, it will send back a RTS rather than a

CTS. This gives it priority to send and empty its buffer. This has to be used carefully; otherwise its usefulness would be negating and lead to many collisions. T-MAC specifies that a node can only use this method if it has lost contention twice to another node.

T-MAC is more energy efficient than S-MAC, but at the cost of throughput and latency. Additionally, it also suffers from S-MAC's scaling problems, in that it incurs a great deal of overhead as the network size increases.

2.1.1.3. B-MAC

Although S-MAC and T-MAC improve the energy limitations of WSN's, they were designed for generic traffic loads. The Berkeley Medium Access Control (B-MAC) [PHC04] protocol, on the other hand, was designed assuming WSN data is sent periodically in short packets. B-MAC is solely a link protocol, requiring other services to be controlled by higher applications. In this manner, the responsibility of optimizing power consumption, latency, throughput, fairness or reliability falls upon the node's applications. Finally, B-MAC adapts more efficiently to a dynamic topology and tolerant of changing network conditions.

B-MAC uses *clear channel assessment* (CCA) to determine if the channel is clear. Using CCA, a node estimates the noise floor by analyzing several signal strength samples of a channel when it is assumed to be free, such as immediately after a packet transmission. When the node is ready to transmit, it monitors the channel's energy and searches for outliers that are significantly below the noise floor. Assuming valid packets would never generate such an outlier, the existence of one proves the channel to be clear. However, if no outliers are discovered after five samples, the channel is presumed to be busy. If the channel is clear, the node will use a random *backoff*, and then run CCA once more. If the channel is busy, the node will again use a random backoff; otherwise it will begin transmitting.

To conserve energy, nodes implement *low power listening* (LPL), whereby nodes cycle through stages and periodically sample the channel. In the first stage, a node is asleep. After being woken by a timer, the node initializes its radio and listens for activity on the channel. If activity is detected, the node remains awake and receives the incoming packet before returning to sleep. If no activity is detected, a timer puts the node to sleep. The interval between LPL samples is maximized to prevent idle listening.

B-MAC exceeds the performance of S-MAC and T-MAC through reconfiguration, feedback and interfaces with higher-layer applications. Further, it does not force applications to incur the overhead of synchronization and state maintenance. With the default B-MAC parameters and no additional information, B-MAC surpasses S-MAC and T-MAC in terms of throughput, latency, and energy consumption [PHC04].

2.1.1.4. PD-MAC

Packets Decision MAC (PD-MAC) [JWZ+08] assumes when a significant event occurs, multiple nodes will sense it and become aware. Under S-MAC, each of these aware nodes would transmit packets, thereby alerting other nodes and producing redundant transmissions that waste the WSN’s energy as well as unnecessarily consume the wireless channel. To address this problem, PD-MAC adds two additional fields to the RTS and CTS packets. The first, *OA*, contains the address of the witness node while the other, *PN*, contains the number of packets.

PD-MAC nodes have the same initiation procedure as S-MAC and form virtual clusters. When nodes witness an event, they compete to transmit by sending a RTS packet. All neighboring nodes within the virtual cluster record the OA and PN fields from the RTS packet and add them to their return CTS packet. If a node receives a RTS packet and also has packets to send, it compares OA fields. If identical, the node discards its packets and immediately goes to sleep, thus preventing a redundant transmission. If the witness nodes in the OA field are different, a node determines if they are neighbors. If so, the PN field is compared to see if the number of packets is similar. A similarly-sized PN field indicates either similar, or the same, data is being transmitted by neighboring nodes. In this case, a node stores the data for future comparison, then goes to sleep. When other packets are received, the node abandons the previously stored packets, or compares the PN field of CTS packets until new data is received, then competes to transmit.

Using PD-MAC, fewer nodes within the WSN transmit, prolonging the network's lifespan. Further, because fewer nodes are transmitting, the wireless medium is less congested, resulting in fewer collisions. PD-MAC reduces average WSN energy consumption by 30% compared to S-MAC [JWZ+08], improves end-to-end delay, and achieves greater delivery accuracy as the density of the WSN increases.

2.1.2. Schedule-Based MAC Protocols

Schedule-based protocols are based upon time-division multiple access (TDMA), using reservations and scheduling to conserve energy. In this manner, they guarantee collision-free communication without contention-introduced overhead by scheduling slots for each node. This also reduces idle listening, resulting in significant energy savings. Using a TDMA protocol, however, requires nodes to form *real* communication clusters rather than the virtual ones found in CSMA protocols. Managing inter-cluster communication and interference is not an easy task. Challenges include determining the slots to be assigned to nodes, high initial overhead to set up and distribute a schedule throughout the WSN, and accurate time synchronization to prevent clock drift so that nodes' time slots do not overlap. Moreover, when the number of nodes within a cluster changes, it is not easy for a TDMA-based protocol to change its schedule without retransmitting overhead packets, thus their scalability is not as good as that of contention-based protocols.

2.1.2.1. TRAMA

TRaffic-Adaptive Medium Access (TRAMA) [ROG06] differs from previously discussed MAC protocols by supporting unicast, broadcast, and multicast traffic. It is inherently collision-free, due to TDMA, and uses a dynamic approach to switch nodes to low power based upon traffic patterns. It consists of three components: the *Neighbor Protocol* (NP), *Schedule Exchange Protocol* (SEP), and *Adaptive Election Algorithm* (AEA). The first two components exchange neighbor information and schedules. The third uses that information to select transmitters and receivers for a time slot, allowing all other nodes to go to sleep, thus achieving collision-free transmissions.

During initialization, TRAMA's NP shares one-hop neighbor information. Each node contends with neighbors to transmit packets containing incremental neighborhood updates in a randomly selected *signaling slot*. In this manner, nodes learn the one-hop neighbors of their one-hop neighbors, thus two-hop neighbor information is propagated across the network. If a node fails to hear from a neighbor after some time, it is removed from that node's neighborhood list. To prevent the premature removal of active nodes, nodes will send signaling packets during its time slot, even if there are no updates.

With two-hop neighbor information known, TRAMA's SEP generates and maintains traffic-based schedule information amongst neighbors. Each node generates its schedule by comparing an interval of slots with its two-hop neighbors. Those slots for which it has highest priority are the slots during which it can transmit. The node announces the neighbors it intends to transmit to by broadcasting a schedule packet containing a bitmap representing each one-hop neighbor. If the corresponding bit in the bitmap is set, that neighbor is an intended receiver. If a transmitting node does not have enough packets to fill its reserved slots, it proclaims so to its neighbors and gives them up for their use. Finally, each node saves its last reserved slot to broadcast its schedule for the next interval. To maintain the schedule, a node's schedule is sent with every data packet. Each schedule has an associated timeout, and nodes are not allowed to change the schedule until this timeout expires, ensuring consistency amongst one-hop neighbors. Each node maintains the schedule of its one-hop neighbors and updates it using the data sent with each data packet. Further, each node listens during a *ChangeOver* slot, the slot after which all reserved slots go unused, to synchronize schedules.

AEA uses neighborhood and schedule information from NP and SEP to select transmitters and receivers for the current time slot, leaving all other nodes to go to sleep and thus achieving collision-free transmissions. Each node executes AEA to decide whether it should transmit, receive, or sleep based upon current node priorities and on the announced schedules from one-hop neighbors. A node will transmit only if it has the highest priority amongst its two-hop neighbors and has data to send. A node receives after it has checked the schedule of the transmitting node and determined it is an intended receiver. Otherwise, the node will sleep. To avoid a hidden node problem, each node must account for the two highest-priority transmitting nodes before going to sleep. Otherwise, a node choosing only the highest transmitting node that does not have packets to send sleeps, while another node that is three-hops away from the other transmitting node, but still within two-hops of the receiving node could transmit as well, thus the receiving node would be asleep and not receive the packet.

TRAMA achieves a 40% higher throughput than S-MAC [ROG06], as well as significant energy savings due to being schedule-based. However, because it is schedule-based, it also incurs an increased delay. As such, it is better suited for applications that are delay tolerant and require reliable delivery guarantees and energy efficiency.

2.1.3. Hybrid Protocols

Hybrid Protocols are a blend of contention-based and schedule-based protocols, using both to achieve energy savings while offsetting their respective weaknesses. Contention-based protocols offer simplicity, flexibility and robustness, and do not require much infrastructure support. These advantages, however, are a result of repeated trial and error and packet collisions can occur within any two-hop neighborhood of a node due to the hidden node problem. These collisions can be minimized using RTS/CTS, however that incurs a high overhead that consumes 40% - 75% of the channel's capacity [RWA+08].

Schedule-based protocols, on the other hand, solve the hidden node problem by scheduling the neighboring nodes to transmit at different times, but suffer from their own disadvantages. Creating an efficient schedule is not easy, and it requires each node maintain clock synchronization. The tighter the synchronization, the higher the overhead required due to more frequent exchanges between nodes. Further, changes to the WSN topology require schedule changes, inducing additional overhead.

2.1.3.1. Z-MAC

Zebra MAC (Z-MAC) [RWA+08] is a hybrid protocol based upon CSMA. It maintains high channel utilization using CSMA under periods of low contention and TDMA under periods of high contention. In its worst case, Z-MAC performs identical to CSMA. It consists of four sequential procedures, *neighbor discovery*, *slot assignment*, *local frame exchange* and *global time synchronization*, which only function during the WSN's initialization period or after significant changes to its topology.

During neighbor discovery, each node periodically broadcasts a ping message, containing an updated list of one-hop neighbors, to its one-hop neighbors. In this manner, each node creates a list of its two-hop neighbors. With this list, Z-MAC uses the DRAND [RWM+06] algorithm to assign each node a time slot, making sure no two-hop neighbors share the same slot. Each node then develops a *time frame*, the period in which it can use its time slot. Ideally, each two-hop neighborhood of nodes shares the same time frame. For a dynamic WSN, however, each topology change would require updated time frames to be propagated throughout the network, wasting energy. To account for topology changes, Z-MAC's *time frame rule* allows each node to maintain its own local time frame that fits its two-hop neighborhood, but avoids conflicting with contending neighbors. After each node has determined its time frame and slot number, it broadcasts them to its two-hop neighborhood and synchronizes their time slots to slot 0. The time slots are maintained by each node periodically sending a synchronization message containing its current clock value.

Z-MAC nodes operate in either a *low contention level* (LCL) or *high contention level* (HCL) mode. While in HCL mode, a node competes to transmit in the current slot only if it owns the slot or is a one-hop neighbor to the owner of the slot. In LCL mode, however, a node competes in any slot. In either mode, the owner of the slot has higher priority over other nodes. If a slot has no owner, or the owner has no data to send, other nodes can use it. A node enters HCL mode when it receives an *explicit contention notification* (ECN) message from a two-hop neighbor within a given time. ECN functions similarly to RTS/CTS, however uses topology and slot information to avoid collisions. A node sends an ECN message when it determines that contention amongst nodes is high by measuring the noise level of the channel.

Z-MAC uses the backoff, CCA and LPL interfaces of B-MAC to implement LCL and HCL. When a node is ready to transmit data, it checks to see if it owns the slot. If it does, it takes a random backoff for a period of time. Once the backoff timer expires, the node uses CCA to sense the channel, and transmits if it is clear. If it is not, it repeats the process until the data is transmitted. If the node does not own the slot and is in LCL, or is in HCL and the slot is not owned by its two-hop neighbors, it takes a random backoff within a contention window and otherwise performs as previously described. If the node does not own the slot and is in HCL because a neighbor sent an ECN, the node sleeps until a slot arrives that it owns or is not owned by a two-hop neighbor, then it wakes up and repeats the previous process. Nodes receive packets using B-MAC's LPL mode.

At low transmission rates, Z-MAC performs no worse than CSMA. As transmission rates increase, however, Z-MAC outperforms B-MAC in terms of throughput, fairness and energy efficiency. Their latency, however, was similar regardless of transmission rates.

2.2. Routing Protocols

Whereas MAC protocols determine when and how nodes communicate with each other, routing protocols direct node traffic in an efficient manner. Adopting the terminology from [BTJ05], an *agent* is defined as a packet responsible for spreading rumors about sensed events in the network, and a *query* as a request packet for receiving information on any event. These two packet types represent the main sources of traffic propagating across a WSN, while each node acts as a router to relay them.

2.2.1. Rumor Routing

The *Rumor Routing* [BE02] protocol improves a nodes' ability to transmit queries and event information throughout a wireless sensor network. The most expedient way to guarantee every query is successful is to flood the WSN with both query and event information. This, however, requires every node to expend energy to receive or transmit the query/event information. Doing so quickly expends energy stores, resulting in nodes expiring quickly and the WSN eventually failing. What's more, each node's memory would quickly fill as it stored query and event information. Furthermore, due to frequent nodes’ transmissions, wasteful collisions would occur frequently. Rumor Routing, on the other hand, conserves energy and memory capacity by selecting a random path for both the query and event information to follow. This reduces the number of transmitting nodes, as well as the number of nodes informed of events, saving energy throughout the WSN. In addition, Rumor Routing provides data redundancy by sharing information throughout the network.

Each node within a WSN with Rumor Routing initializes using an active broadcast to locate neighboring nodes. These neighbors are added to a list within the node’s memory, which is maintained through subsequent active broadcasts, or by passively listening to other nodes’ broadcasts. Additionally, each node maintains an event table containing forwarding information for each event it has been informed of.

If a node witnesses an event, it adds it to its event table and generates an agent. The agent traverses the network, “informing” other nodes of events it has witnessed. The agent uses a straightening algorithm to maintain a straight path, thereby transmitting information as far across the network as possible. The straightening algorithm uses a list of current neighbors and compares it to a list of previously visited nodes. Prior to transmitting, a node chooses a neighbor the agent has not previously visited. In this way, agents follow a fairly “straight” path, eliminating the possibility of the transmission being sent repeatedly to nodes that have already received it.

The agent contains a list of witnessed events as well as the number of hops to each event. When received by a node, the agent synchronizes its list with the node’s list so both of their tables contain routes to every event. In addition, since agents are broadcast in the WSN, every neighboring node within receiving distance of the agent receives the updated information and updates their event tables as well. This results in a "thick" path of updated nodes. This behavior continues until the agent's lifetime expires.

To receive event information, a node within the WSN generates a query. The query is sent in a random direction to a neighboring node. That node, if aware of a route to the event, forwards the query accordingly. Otherwise, it forwards the query in a random direction to one of its neighboring nodes. The query uses the same algorithm as the agent to determine the direction to send the query, thus avoiding the same nodes. Should a node within the network fail, however, it is possible the query could be caught in a loop. To avoid this, each query is assigned a limited lifetime, as well as a random identification number. If a query arrives at a node which has already forwarded it, the node instead sends the query to a random neighbor, thus breaking the loop. This process continues until the query has reached a node that has information about the event, or until the query's lifetime expires. If the originating node of a query determines it did not reach the event, it can retransmit the query, quit the query, or flood the network with the query.

The Rumor Routing protocol has several drawbacks. First, its straightening algorithm is not always effective in ensuring agents and queries are spread across the network. Although it prevents revisiting nodes and loops, it is susceptible to following a spiral pattern. Thus, the agent or query could stay within a relatively small area within the WSN, reducing the probability of a successful query. Furthermore, when dealing with a large WSN, the agent's and query's list of visited nodes grows each time they are forwarded. Eventually, this information constitutes an enormous amount of data, requiring each node to expend a greater amount of energy with each subsequent transmission, resulting in earlier network failure.

2.2.2. Zonal Rumor Routing (ZRR)

Another limitation of Rumor Routing is the next node a query or agent visits is randomly selected. Each of the neighboring nodes, near or far, have an equal probability of being selected. If nearby nodes are chosen more often than distant nodes, queries and agents are more likely to remain within a small area and take longer to intercept one another. If distant nodes are selected, however, transmissions are further from the original node, allowing the agent to spread information to more of the network in less time. *Zonal Rumor Routing* [BTJ05] is an extension of the Rumor Routing protocol, allowing agents and queries to spread across the WSN with greater efficiency. The network is partitioned into zones, with each node being a member of one zone. Unlike Rumor Routing, where the query or agent randomly selects an unvisited neighboring node as the next hop, queries and agents using Zonal Rumor Routing randomly select a node from an unvisited neighboring zone.

As with Rumor Routing, every node in Zonal Rumor Routing maintains a list of its neighboring nodes, their distance, and a list of events the node has witnessed or learned of. Unlike Rumor Routing, however, each node also maintains a list of each neighboring node's particular zone. Each node has a certain probability of being selected a zone leader. When the network initiates, zone leaders broadcast a message to neighboring nodes, asking them to join their zone. If a node is already a member of another zone, it responds with its unique node id and zone id, which the zone leader uses to update its neighbor list. All other nodes ignore this broadcast. If a node is not already a member of a zone, it joins that zone and forwards the request to its neighboring nodes. Upon receiving their reply, it updates its neighbor list. The zone leader, having heard the forwarded broadcasts, updates its table with the new nodes. This process continues until all nodes have joined a zone and all requests stop. At this point the network is stabilized and each node is aware of their zone membership, and that of their neighbors.

The routing algorithm for agents and queries is similar to that of Rumor Routing. The difference, however, is each agent and query also maintains a history of visited zones, beginning with the zone it originated from. When deciding the next hop, the agent or query uses its list to find a neighboring node from a different zone. As in Rumor Routing, the agent or query shares its event table with the node, and all neighboring nodes within broadcast range of an agent also update their tables. If the agent or query is unable to find a neighboring node from a different zone, it randomly selects a neighboring node.

Because the objective of Zonal Rumor Routing is to spread the agent or query as far as possible across the network, the goal is to choose the furthest neighboring node as the next hop. Should the number of zones be near or equal to the number of nodes, however, the protocol effectively acts the same as the Rumor Routing protocol. With an optimal number of zones, agents and queries will reach a wider region of the sensor network with fewer transmissions, increasing the probability of a successful query and reducing the total energy consumption of the network.

2.2.3. Straight Line Routing (SLR)

The two previously discussed protocols could be classified as random-walk protocols. Although they use an algorithm to travel in a “straight” path, and prevent backtracking to nodes previously visited, they have the potential to take inefficient paths. Thus, more transmissions are required. This results in a greater delay for a successful query and thus more energy consumed. Additionally, both protocols' agents and queries maintain lists of visited nodes and zones. When each is forwarded, this list grows larger and causes each subsequent node to incur a greater transmission time, thus expending more energy.

The *Straight Line Routing* [CSC05] protocol addresses these problems by keeping both the agent and query transmission paths as straight as possible. As with the previous two protocols, Straight Line Routing chooses its path one hop at a time. Ideally, each future node lies along the desired trajectory, at the furthest reach of the node's transmission range. Since this is not always possible, Straight Line Routing selects the next node from a section of the current node's transmission range called the *Candidate Region*.

The *Candidate Region* is an overlapping region of two parameters: the *Outside Band* and *Inside Band*. The outside band is formed by the radius of the node previously visited by an agent or query, where the distance is determined by its furthest transmission range. The inside band is formed by the radius of the node in which the agent or query currently resides. This radius can be adjusted depending upon the size of the WSN, but is typically half the furthest transmission range of the current node.

To determine the candidate region, each node maintains two variables: *FlagIn* and *FlagOut*. Straight Line Routing assumes the sending node can be identified, and calculates the distance between the receiving and sending node based on its signal strength. Using the distance from the previous node, and the distance from the current node, a node can determine if it is within the candidate region. If the node is within the inside band of the current node, it will set its *FlagIn*. If it is within the outside band, it sets *FlagOut*. If both flags are set, the node is considered as a potential next hop.

Once determined to be in the candidate region, a node starts a timer equal to the sum of the inverse of both the distance of the outside band and inside band. In this way, the furthest node's timer will expire first. Once the timer expires, the node sends a message to the transmitting node, designating it the next hop. Other nodes within the candidate region will receive the transmission and stop competing.

Drawbacks to this protocol include nodes competing to be the next hop must receive two transmissions to determine whether or not they are in the candidate region, using twice the energy and decreasing the probability of success by half. Additionally, the furthest distance of the next hop is limited by the radius of the inside band. Assuming this distance is half the radius of the current node's transmission range, the number of hops an agent or query must make is twice that of other protocols. This increase in hops increases delay for queries.

2.2.4. Trajectory-based Selective Broadcast Query (TSBQ)

Unlike the other protocols, the *Trajectory-based Selective Broadcast Query* [MBK+07] protocol minimizes a network's total energy expenditure by determining an optimum number of transmissions, or time to live (TTL) for each agent. Thus it accounts for the energy expended to inform a WSN, as well as simultaneously taking advantage of the broadcast feature of wireless to query multiple neighboring nodes at once.

TSBQ generates an agent upon witnessing an event. That agent is forwarded using a straight-line trajectory to a single node using most forward routing to eliminate looping. Thus, the number of informed nodes is minimized, reducing the amount of data transmitted. Additionally, the informed nodes are spread across great distances, reducing the probability of a large number of informed nodes within small areas of the network. If a node cannot forward the agent in the direction of the desired trajectory, it randomly chooses a new trajectory. To conserve energy throughout the WSN, all nodes within reception range of the transmitting node, but not selected as the next hop, deactivate their receiving hardware according to the TDMA MAC protocol, where transmitting and receiving nodes coordinate during the MAC protocol's initialization period. When a node receives the agent from a transmitting node, it makes an entry in its event table to include the type of data advertised, the location of the witness node, and a copy of the data. This process continues until αN nodes have been informed, where N is the number of nodes in the network, and α is chosen from $\{1/N, 2/N, ..., (N-1)/N\}$. After αN nodes have been informed, the agent is terminated.

With TSBQ, a node needing access to services or data generates a query in a random direction. Similar to the agent transmissions, queries are forwarded along straight-line trajectories, but are also broadcast to a subset of its neighboring nodes closer to it than the next potential hop. By staying in a straight line, the probability the currently transmitting node's neighbors have not already been queried increases. Again, via the TDMA MAC protocol, those nodes not selected to receive the transmission deactivate their receivers to conserve energy. If the querying node's neighbors are not informed of the desired event, it selects one of its one-hop neighbors along the desired trajectory as the next querying node. Using most forward routing, this newly selected querying node selects a new query node along the desired trajectory, and queries a subset of its neighbors closer than the new query node. If none of the queried nodes are informed, the query is forwarded to the newly selected query node. This process repeats until the query is successful or terminated. If successful, the current querying node forwards the desired information to the original query node using most forward routing back along the trajectory defined by the current query node and the original query node.

2.3. Summary

This chapter provides a review of literature that is relevant to this research. It discusses how medium access control protocols are responsible for regulating how a medium is shared, ensuring no two nodes in a WSN interfere with each other's transmissions. Contention-based, schedule-based, and hybrid MACs are examined and their performance is compared. Routing protocols, responsible for directing node traffic in an efficient manner, are also discussed. Lastly, agents and queries are identified and defined.

3. Methodology

3.1. Problem Definition

There are two main features that set wireless sensor networks apart from traditional ad hoc networks, the first being size. While ad hoc networks may contain tens or hundreds of nodes, WSNs are anticipated to be most effective as high-density networks composed of nodes ranging in scale from thousands to millions. The other differentiating feature is a node's power supply. For most ad hoc nodes, power is not an issue. They are either connected directly through a power line or cable, or operate on batteries that can be recharged. WSN nodes, however, are likely to be isolated after deployment, and function only as long as their internal battery lasts.

The TSBQ protocol was designed specifically for wireless sensor networks with these unique features in mind. Unlike previous protocols, TSBQ minimizes a network's total energy expenditure by setting an appropriate lifetime for each agent and query, as well as limiting the number of times each agent can be transmitted. During TSBQ's development, a simulation model was created [MBK+08] to examine the performance of a rumor routing search protocol modified with TSBQ's unique agent/query limitations applied. The model measured the mean rate of packet arrivals per node to estimate the energy expenditure of the network, as well as the total proportion of query failures to determine its effectiveness.

However, the effect each of the protocol's parameters had upon the network's performance was not thoroughly analyzed. The simulation model was based upon an analytic model that assumed all packets arrive according to a Poisson process, thus all packet-related parameter distributions were exponential. To expand upon this model, and aid in the verification of future analytic models, different distributions are applied to some of the packet-related parameters, and the effect each parameter has upon the network performance is analyzed.

3.1.1. Research Goals

Goals for this research are to:

1. Update the modified rumor routing search protocol [MBK+08] model so that its packet-related parameters can specify any distribution and verify its performance.
2. Analyze the effect each parameter has upon the performance of a WSN, focusing specifically on the mean rate of arrivals per node, total proportion of query failures throughout the network, and the mean proportion of time each node is uninformed.

3.1.2. Approach

The first goal of this research requires the modified rumor routing protocol in [MBK+08] to accept any distribution as an input to its packet-related parameters. The performance of the updated protocol is verified against the original, with any differences explained and justified. It is vital the updated protocol perform the same to ensure the accuracy of any comparisons between the two models, as well for use in future research.

The second goal is accomplished by adjusting the parameter distributions of the protocol, as seen later in Table 2, and analyzing their effect upon the system. The mean rate of arrivals per node is an energy-focused metric that estimates the total energy expenditure of the network, while the proportion of query failures determines the protocol's effectiveness at answering queries. The mean proportion of time a node is uninformed is measured to aid in the development and verification of future analytic models.

3.2. System Boundaries

The system under test (SUT) is the wireless sensor network, while the components under test (CUT) are the nodes in the WSN and the updated modified rumor routing protocol. The system is thus named the modified rumor routing protocol, or *MORRP*. A diagram of MORRP is shown in Figure 2. The system services, workload, performance metrics, parameters, factors, and responses are discussed in later sections.

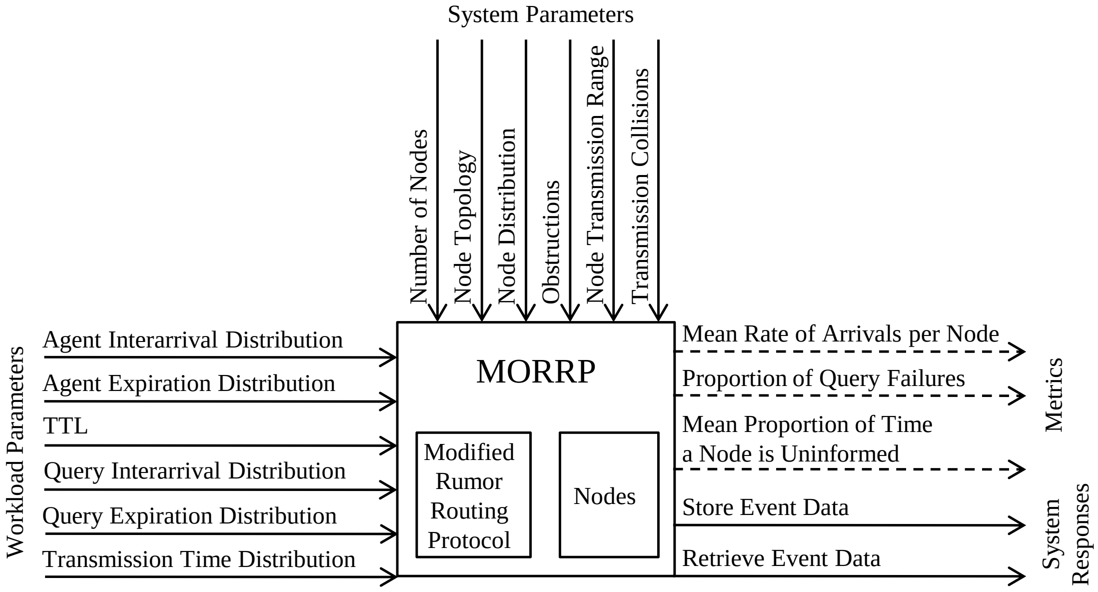

Figure 2: The modified rumor routing protocol system

3.3. System Services

The primary function of a wireless sensor network is to monitor an environment, sense events, and relay event data to answer queries. In addition to transmitting and receiving agents or queries, WSN nodes perform various other functions to include initializing and maintaining localization information, synchronizing transmission periods, and computing data. These functions, however, are primarily a result of a node's MAC protocol and hardware. As such, the system analysis focuses only on the services provided by the modified rumor routing protocol. In the case of MORRP, these services can generally be described as storing sensed event data and locating sensed event data. These services and their possible outcomes are summarized in Table 1.

3.4. Workload

The workload of a wireless sensor network, when considering energy efficiency, is a function of the time every node spends in a particular state: sleeping, computing, sensing, receiving, or transmitting. The amount of energy used while sleeping or computing is insignificant compared to the energy a node expends while transmitting or receiving [ROG06, TAH02], thus it is not considered further. Additionally, the time a node spends sensing for events is a function predetermined by the user, not the search protocol, and is also excluded. Therefore, the workload for the modified rumor routing search protocol is a result of node transmissions and receptions.

Table 1: System services and possible outcomes

Service	*Possible Outcomes*
Store sensed event data	Event data correctly stored
	Event data stored with errors
	Event data not stored
Locate sensed event data	Event data located
	Event data not located

3.3.1. Workload parameters

The parameters that affect the MORRP workload include:

- The time between sensed events/generated agents (*agent interarrival distribution*)
- An agent’s lifetime (*agent expiration distribution*)
- The number of times an agent can be transmitted (TTL)
- The time between queries generated (*query interarrival distribution*)
- A query’s lifetime (*query expiration distribution*)
- The time between agent/query transmissions (*transmission time distribution*)

The agent interarrival distribution and query interarrival distribution parameters are responsible for the number of agents and queries generated within the system. Their expiration distributions, however, limit the time each has to traverse the network, in turn limiting the number of transmissions and receptions. Likewise, the TTL parameter limits the number of nodes an agent may visit, also limiting the number of agent transmissions and receptions. The transmission time distribution parameter is responsible for the time a node needs to process and transmit an agent or query. If this parameter is too large, nodes' transmission queues could overflow, resulting in the failure of the protocol.

3.5. Performance Metrics

Three metrics are used to evaluate the MORRP performance:

- The mean rate of packet arrivals per node
- The proportion of query failures
- The mean proportion of time a node is uninformed

Transmitting and receiving consumes the majority of a node's energy [ROG06, TAH02]. The mean rate of arrivals per node accounts for the average rate both agents and queries are received by a given node within the network. As specified by the modified rumor routing protocol, a node transmits agents/queries to a single neighbor in a unicast manner. As a result, every packet received by a node is equivalent to a single transmission by a neighbor. By measuring the rate of arrivals, a node's energy consumption can be estimated, which in turn can assist in determining the total network energy expenditure. The goal with this metric, therefore, is to minimize the rate at which agents and queries are received by each node, thus reducing the networks total energy consumption. Reducing the rate of arrivals too much, however, can result in the network failing to answer queries in a timely manner.

The protocol's level of success is determined by measuring the proportion of queries that fail. A *query failure* is defined as a query that expires in a node's transmission queue prior to locating an informed node. If a significant proportion of queries are unanswered, the network is failing in its primary services. As a result, the proportion of query failures must remain less than the user's specified threshold.

The mean proportion of time a node is uninformed is used in the development of future analytic models that use various distributions for the workload parameters.

3.6. System Parameters

Parameters affecting the MORRP performance are:

- The number of nodes in the WSN
- The node distribution
- The node topology
- Obstructions within the network
- Individual node transmission range
- The probability of transmission collisions

The number of nodes in a WSN, assuming a static deployment area, will affect the number of neighbors each node has. Denser networks provide additional neighbors a node can transmit to. This lessens the probability of a node receiving a packet, thus extending its lifetime. Similarly, the distribution of nodes affects how many neighbors a node will have. In a uniformly distributed network, each node will have an equal number of neighbors. A randomly distributed network, however, could result in a node having a single neighbor to communicate with, thus shortening its lifetime. A node's transmission range also determines how many neighbors a node has. A greater range, however, requires more energy per transmission.

The topology of the network, as well as any obstructions within the network, also affects how often a node will be required to transmit and receive. For instance, if the topology contained a bottleneck, nodes residing in the bottleneck will forward packets between the two sides of the network. These nodes will consume their limited power much sooner than the other nodes in the network, resulting in premature failure and segregating the network.

The time required transmitting an agent or query, and the TTL of an agent or query affects how much energy is spent by nodes in the WSN. The longer it takes to transmit, the more energy is expended. Similarly, the longer the TTL of an agent or query is, the more nodes they can hop to, using more energy. In addition, the retransmission of an agent or query requires additional energy to be expended to ensure the data is forwarded.

3.7. Factors

The seven factors used to evaluate the protocol are listed in Table 2. To remain consistent with [MBK+08], the values for each factor are similar. The first factor is the number of nodes within the WSN. A successful WSN protocol must scale, therefore this factor will be evaluated at levels of 500 and 5,000 nodes. Increasing the number of nodes within the network is expected to increase the mean rate of arrivals per node, due to the probability of each node having more neighbors. The proportion of query failures is expected to remain relatively stable, for although the number of agents generated will increase with the additional nodes, so too will the number of queries.

Table 2: Factor and levels for the MORRAS simulation

Factors	*Levels*
N	500
	5000
Agent Interarrival Distribution	Exponential: rate = 0.005
	Exponential: rate = 0.01
Agent Expiration Distribution	Exponential: rate = 0.3
	Uniform: a = 0, b = 6.67
Query Interarrival Distribution	Exponential: rate = 0.04
	Exponential: rate = 0.06
Query Expiration Distribution	Exponential: rate = 0.5
	Uniform: a = 0.01, b = 3.99
Transmission Time Distribution	Exponential: rate = 0.2
	Rayleigh: scale = 0.39894
	Uniform: a = 0.01, b = 0.99
TTL	5
	15
	25

The agent interarrival distribution parameter determines the time between generated agents, each representing a sensed event, by a single node. This factor is evaluated using two levels, both exponential distributions. The first is exponential with a rate of 0.005 agents/second/node or a mean of one agent generated every 200 seconds per node. The second level is exponential with a rate of 0.01 agents/second/node, equating to one agent generated every 100 seconds per node.

The agent expiration distribution is a factor controlled by the user and determines the agent's lifetime. A longer agent lifetime allows a node to travel further within the network, informing additional nodes and increasing the probability of queries being answered. However, it also results in additional transmissions and receptions, thus causing nodes to expend more energy. This factor is evaluated for two levels, both with a mean agent lifetime of 3.3333 seconds. The first level is an exponential distribution with a rate of 0.3, and the second is a uniform distribution with a minimum value of 0 and maximum of 6.67.

The time between queries generated is determined by the query interarrival rate. This factor is controllable by the user and is evaluated using two levels, both exponential distributions. The first is exponential with a rate of 0.04, or one query generated every 25 seconds per node, and the second is exponential with a rate of 0.06, equating to a mean of one query generated every 16.6666 seconds per node.

The lifetime of the query is determined by the query expiration distribution. As with the agent expiration distribution, increasing this factor allows a query to persist in the network longer, thus increasing the likelihood of it discovering an informed node. However, with each additional transmission and reception, nodes must expend additional energy. This factor is evaluated for two levels, both with a mean of 2 seconds. The first is an exponential distribution with a rate of 0.5, and the second is a uniform distribution with a minimum value of 0.01 and maximum value of 3.99.

The transmission time distribution is the time a node requires to process and transmit an agent or query. A longer transmission time increases the likelihood of a node's transmission queue becoming backlogged. In addition, although not monitored in this simulation, it will increase the latency of a successful query. This factor is evaluated using three levels, each with a mean of 0.5 seconds. The first is an exponential distribution with a rate of 2, the second a Rayleigh distribution with a scale of 0.39894, and the third a uniform distribution with a minimum value of 0.01 and maximum value of 0.99.

The final factor considered is the TTL of an agent; queries are unaffected by this factor. Unlike the agent expiration distribution, which sets the lifetime of an agent, the TTL factor determines the number of times an agent can be transmitted to a neighboring node before expiring. It is assumed a node will be successfully transmitted as many times as the TTL factor allows before its lifetime expires. By increasing the TTL, additional nodes are informed by an agent, which increases the likelihood of a query discovering an informed node. However, it also increases the number of transmissions and receptions required to transmit an agent, resulting in increased energy expenditure. In [MBK+08], the greatest change in network performance occurred for TTL < 26. In the interest of time, as each 5000-node simulation takes hours to complete, the TTL factor is evaluated using levels of 5, 15 and 25.

3.8. Evaluation Technique

The protocol is evaluated using OPNET Modeler 15.0 on a Linux computer running CentOS 5 with four AMD 64-bit processors. There are presently no physical WSNs in existence with the number of nodes required to model the protocol. Additionally, the analytic equation for the protocol, developed in [MBK+08], assumed exponential distributions for each workload factor. As a result, there is no data to compare the results from this simulation model with.

3.9. Experimental Design

The time to complete a simulation using a single set of parameters with a 500 node network is approximately three seconds of real time. A 5000 node network, on the other hand, requires approximately three hours of real time. Neither of these times is exceptionally large, thus a full-factorial experimental design is used. To ensure the simulation's performance is constant, each set of factors is simulated three times.

3.10. Summary

This chapter describes the research goals and hypothesis for this thesis, as well as the approach to achieve those goals. It identifies and justifies the system and its components, as well as the system services, workload, performance metrics, parameters, factors and levels. Finally, a simulation model is described and justified as the means to evaluate the effect each parameter has upon the protocol.

4. Results

4.1. Node Model

OPNET Modeler 15.0 is used to evaluate the effect each factor has upon the performance of the protocol. Each node is modeled in OPNET as a wireless transceiver, as shown in Figure 3, with a fixed transmission range. Sensed events are simulated using a processor module, the *agent generator*, which generates an agent for each simulated event according to the agent interarrival distribution parameter set by the user. Each agent is forwarded to the transmission queue module to await transmission to a random neighbor, while a copy is stored in the *event table* queue module. An agent will remain in the event table until its lifetime, determined by the agent expiration distribution parameter, expires. In this manner, the event table resembles a G/G/∞ queue. If at least one agent is present within the event table, the node is considered informed and capable of answering any query.

Queries are also generated by a processor module, the *query generator*, according to the query interarrival distribution parameter set by the user. Once a query is created, the node checks the local event table. If an agent is present, the query is “answered” locally and proceeds no further. Otherwise, the query is forwarded to the transmission queue. If a query expires while awaiting transmission, it is a query failure.

Packets received from neighboring nodes must first pass through a *splitter*, which ensures a copy of all agents are forwarded to the event table, before being sent to the transmission queue. The splitter has no affect upon queries, other than to forward them to

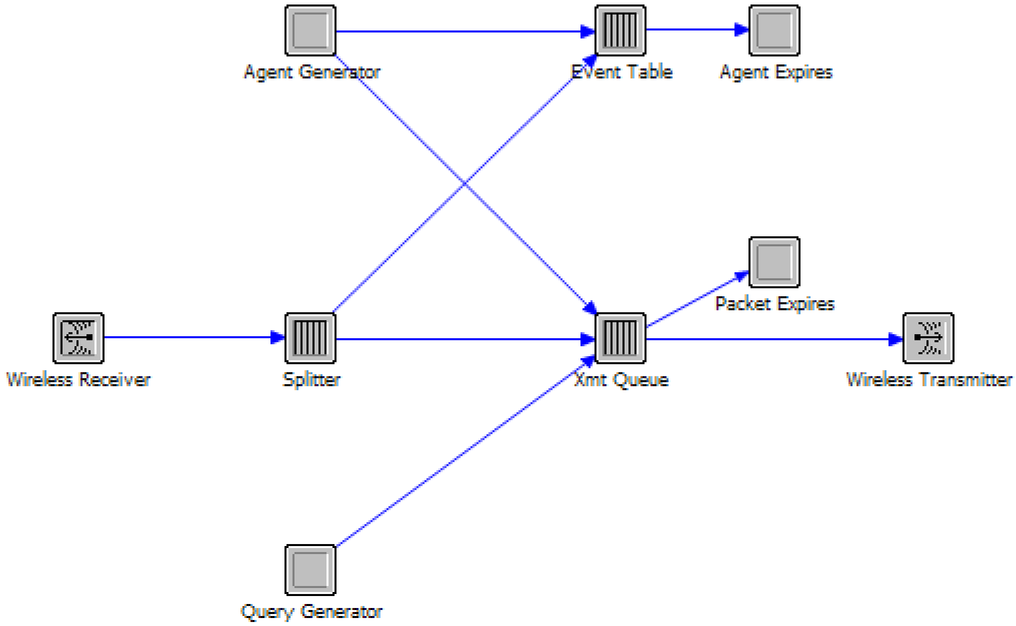

Figure 3: A MORRP node modeled in OPNET

the transmission queue. Due to its simple nature, the splitter adds no delay to the time an agent or query spends within a node.

All packets arriving at the transmission queue are scheduled for transmission according to a first in, first out service discipline and are serviced at a speed determined by the transmission time distribution parameter set by the user. If an agent or query expires prior to being transmitted, it is removed from the transmission queue. Thus, the transmission queue is a G/G/1 queue with reneging.

When an agent enters the transmission queue, a node will determine if its TTL counter has expired. If so, the agent is removed from the queue and deleted. A copy of the agent, however, will remain in the node's event table until its lifetime expires.

Otherwise, the agent's TTL counter is decremented and it remains in the queue until either its lifetime expires or it is transmitted to a random neighbor within the node's transmission range.

When a query enters the transmission queue, the node checks its local event table for any agents. If an agent is present, the query is answered and need not be transmitted further, thus it is removed from the transmission queue and deleted. If no agent is present, the query remains in the transmission queue until its lifetime expires or it is transmitted to a random neighbor within the node's transmission range.

Every node in the simulated network is identical, both in design and configuration. Parameters for each module that can be configured by the user prior to running the simulation are shown in Table 3.

Table 3: User-adjustable simulation parameters.

Module	*Parameter*	*Description*
Agent Generator	TTL	The maximum number of times an agent may be transmitted
	Agent Interarrival Distribution	The mean time between agents generated by a node
	Agent Expiration Distribution	The mean lifetime assigned to an agent upon its generation
Query Generator	Query Interarrival Distribution	The mean time between queries generated by a node
	Query Expiration Distribution	The mean lifetime assigned to a query upon its generation
Transmission Queue	Transmission Time Distribution	The mean time required to process and successfully transmit an agent/query to a neighboring node

4.2. Metrics

There are three indicators of network performance measured during the simulation: mean rate of packet arrivals per node, proportion of query failures, and mean proportion of time a node is uninformed. These metrics, however, are only collected after the simulation has reached steady state. The *measured time* is thus the time during which metrics are collected, equating to the duration of the simulation minus the time required to reach steady state. In [MBK+08], 60 seconds was deemed a sufficient time for the network to reach steady state, and is used for each simulation in this thesis.

The mean rate of packet arrivals per node, *MRPAN*, is an indicator of the network's total energy expenditure. It is

$$MRPAN = \left(\sum_{i=1}^{N} Packet\ Arrivals_i\right) \Big/ t_{measured} \Big/ N \qquad (1)$$

where N is the number of nodes in the network, and $t_{measured}$ is the measured time.

The proportion of query failures, *PQF*, is an indication of the modified rumor routing protocol's ability to successfully answer queries, or

$$PQF = \frac{\sum_i^N Query\ Failures_i}{\sum_i^N Queries\ Created_i - \sum_i^N Stranded\ Queries_i} \qquad (2)$$

where N is the number of nodes in the network and a *stranded query* is a query that remained in a node's transmission queue as the simulation ended. Stranded queries cannot be counted in the proportion of query failures because they did not have a chance to succeed or fail.

The mean proportion of time a node is uninformed, *MPTNU*, is an important component for developing future analytic models. This metric is the total time each node is uninformed during the measured time, divided by the number of nodes within the network, or

$$MPTNU = \sum_{i=1}^{N} \left(\frac{Time\ Uninformed_i}{t_{measured}} \right) \Big/ N \tag{3}$$

where N is the number of nodes in the network, and $t_{measured}$ is the measured time.

4.3. Model Verification

To verify updates made to the original OPNET code [MBK+08] did not alter the modified rumor routing protocol's performance, an identical copy of the original network configuration, in which a thousand nodes were randomly dispersed throughout a 3335m x 3335m area, was created using a scenario duplication feature in OPNET. This procedure ensured every node in the duplicated scenario was in the exact same location as the original scenario. Updates were only made to the duplicated scenario to maintain the integrity of the original scenario, thus any differences between the two would be a direct result of the updated code. The nodes within the duplicated scenario used the updated version of the modified rumor routing protocol. The parameters for both scenarios were identical and are in Table 4.

In [MBK+08], it was determined that a warm-up period of 60 seconds was sufficient for the network to reach a steady state, and that results obtained after a simulation time of 900 seconds were statistically indistinguishable from results using longer times, i.e., several hours. As such, all verification simulation trials were conducted using a simulation time of 900 seconds, with no performance data collected until after the steady state time of 60 seconds had been reached. Individual simulation trials were

Table 4: Parameters for OPNET model verification simulations

Parameter	*Distribution*	*Mean*
Nodes	Constant	1000
Deployment Area	Constant	3335m x 3335m
Transmission Range	Constant	133m
Agent Interarrival Distribution	Exponential	200 sec
Agent Expiration Distribution	Exponential	3.3333 sec
Query Interarrival Distribution	Exponential	20 sec
Query Expiration Distribution	Exponential	2 sec
Transmission Time Distribution	Constant	0.2 sec

conducted for each TTL value ranging from 1-25, three replicates each, resulting in a total of 75 trials. The mean rate of arrivals per node and the proportion of query failures in the network, with 95% confidence intervals, are shown in Figure 4 and Figure 5. The original code did not measure the mean proportion of time a node was uninformed, thus there was no way to verify this metric with the original code.

The x-axis in Figure 4 and Figure 5, α, is the proportion of network nodes informed by an agent, and is directly correlated to the network's TTL parameter; $\alpha = (TTL+1)/N$.

The results of the trials indicate the updated OPNET code's performance is nearly identical to that of the original code [MBK+08], but not identical as one would expect. This is most likely a result of the original code having been simulated in 2007 using OPNET version 10.5, while the updated code used was simulated using OPNET version 15.0. Although identical seed values were used for both sets of trials, it is reasonable to

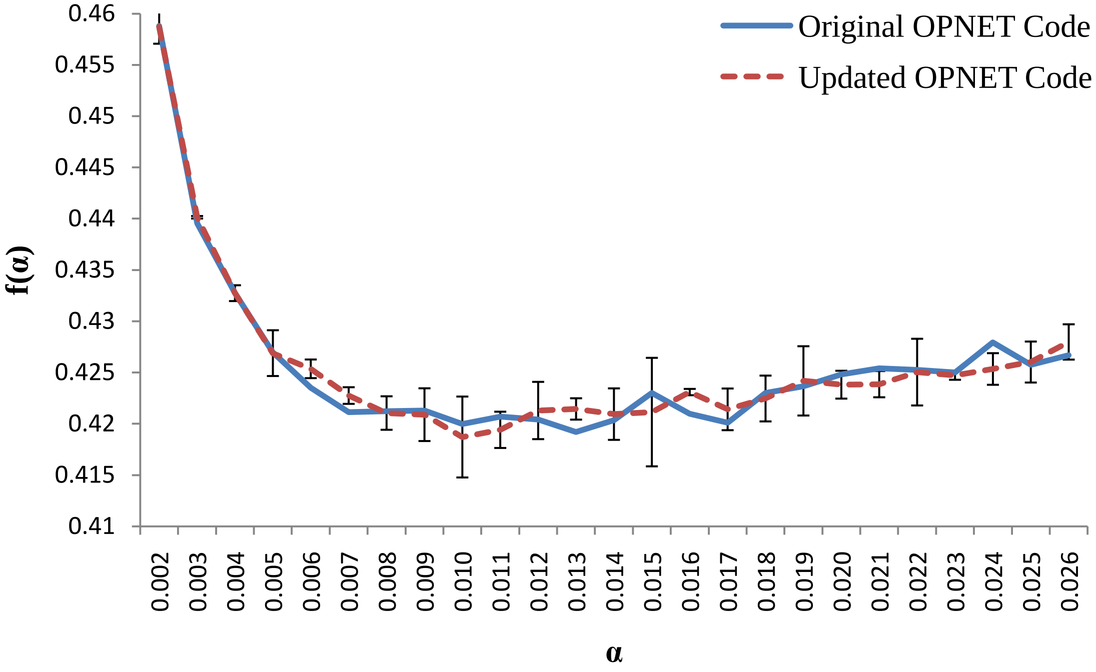

Figure 4: Mean rate of arrivals per node as a function of alpha

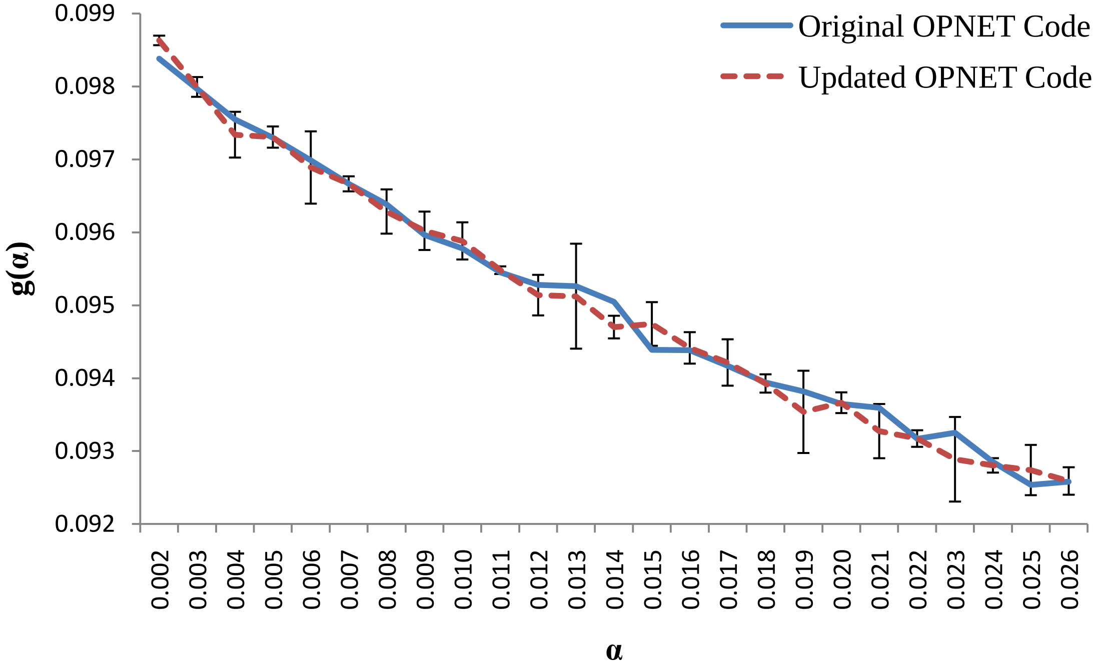

Figure 5: Proportion of query failures as a function of alpha

assume changes between the two OPNET versions resulted in different random number generators associated with the seed values. Based on these results, it is concluded the updated OPNET code's performance is sufficiently similar to the original code to proceed.

4.4. Simulation Results

Two separate network configurations were created within OPNET: one with 500 nodes, the other with 5000. The nodes were distributed randomly within a 3335m x 3335m area using the random disbursement feature in OPNET. Once placed, their location remained static for the duration of every trial. As with the 1000-node verification simulation, each network was given 60 seconds to reach steady state before data was collected, and each simulation trial's duration was 900 seconds. Additionally, three replicates were conducted for each trial, resulting in a total of 864 trials.

A balanced analysis of variance (ANOVA) was calculated from the trials for each of the three performance metrics using the program Minitab. Each ANOVA was initially calculated using every factor and combination of factor interactions to evaluate the effect each had upon the metric. Factors and interactions that proved statistically insignificant, i.e., having a p-value > 0.05, were removed from the model and the ANOVA was recalculated. From the resulting tables, factors and interactions whose effects were inconsequentially small, despite being statistically significant, were also removed from the model and each ANOVA was recalculated. Thus, the resulting ANOVA tables for each metric contain only statistically significant factors and their interactions that had a reasonable effect upon the metric. These tables are in Appendix B, and their residual plots are shown in Figure 6, Figure 8, and Figure 10. Each metric is discussed separately below.

4.4.1. Mean Rate of Arrivals per Node

In an ANOVA model, residuals are assumed to be normal and independent with a constant variation. Residual plots of an ANOVA model are a useful tool in verifying these assumptions. In Figure 6, the histogram indicates the residuals follow a normal distribution curve. The normal probability plot shows the residuals are linear, with the exception of a few outliers in the tail, also indicating the residuals follow a normal distribution. From these two plots, the normality of the residuals is verified.

The outliers in the residual plots, having a positive or negative residual value greater than 0.0035, are listed in Appendix C. Of these 24 outliers, all but two are associated with the factor N = 500. Residuals are the difference between the observed and predicted responses of the model. Because the 5000-node model has 10 times more nodes than the 500-node model, there is more data to sample, thus it is assumed there would be less error in a larger network. Still, the value of the outlier with the greatest residual is 0.0055696, which is extremely small.

No visual trends are present within the residual versus fits and residual versus order plots, thus the independence of the residuals is verified. In addition, the spread of residuals in the residual versus fits plot is fairly stable, verifying that the residuals have a constant variation. With the ANOVA assumptions verified, the ANOVA is an appropriate tool and the factors and interactions affecting the metric are analyzed.

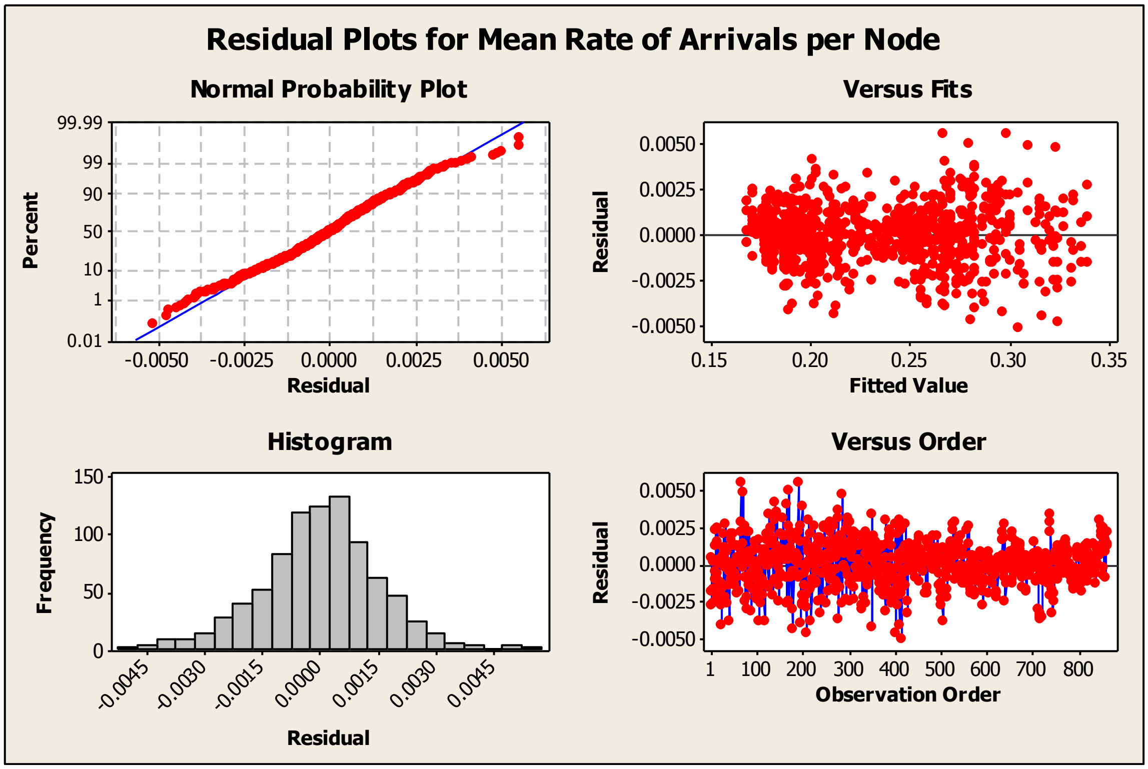

Figure 6: ANOVA residual plots for the mean rate of arrivals per node

The ANOVA table for the mean rate of arrivals per node is found in Appendix B. Comparing the sum of squares value for each factor and interaction of factors with the total sum of squares value reveals that approximately 93% of the total variation is explained by three factors, shown in the abbreviated Table 5.

The factor having the greatest effect on the mean rate of arrivals per node is the query interarrival distribution, accounting for 73.64% of the variation. This is not surprising, since this factor directs each node to generate a query an average of once every 16.7 or 25 seconds, depending upon the factor level. In comparison, agents, which account for all the other packets in the network, are generated by each node once every 100 or 200 seconds. As a result, the majority of the packets being received by a node are

Table 5: Factors with the main effect on the mean rate of arrivals per node

Source	*DF*	*SS*	*MS*	*F*	*P*
Query Interarrival Dist.	1	1.171213	1.171213	464431.36	0.000
Transmission Time Dist.	2	0.176692	0.088346	35032.58	0.000
N	1	0.121273	0.121273	48089.45	0.000
…	…	…	…	…	…
Total	863	1.590559			

going to be queries. In general, an exponential query interarrival distribution with a rate of 0.4 resulted in a lower mean rate of arrivals per node (μ = 0.200, σ = 0.019) than an exponential query interarrival distribution with a rate of 0.6 (μ = 0.270, σ = 0.020).

The next factor with the most effect on the metric is the transmission time distribution, explaining 11.11% of the total variation. This factor is responsible for how quickly both queries and agents are transmitted, and thus received, by nodes. For this factor, the uniform transmission time distribution with a minimum time of 0.01 seconds and a maximum time of 0.99 seconds resulted in a lower mean rate of arrivals per node (μ = 0.223, σ = 0.037) than an exponential distribution with a rate of 2 (μ = 0.232, σ = 0.039) or a Rayleigh distribution with a scale of 0.39894 (μ = 0.257, σ = 0.045). Consider the cumulative distribution functions of all three levels, shown in Figure 7. The probability of the Rayleigh distribution having a transmission time less than or equal to the mean of 0.5 seconds is 79.2%, compared to 63.2% for the exponential distribution and 50.5% for the uniform distribution. With a lower transmission time, packets will spend less time in a node’s queue and arrive at a higher rate to neighboring nodes.

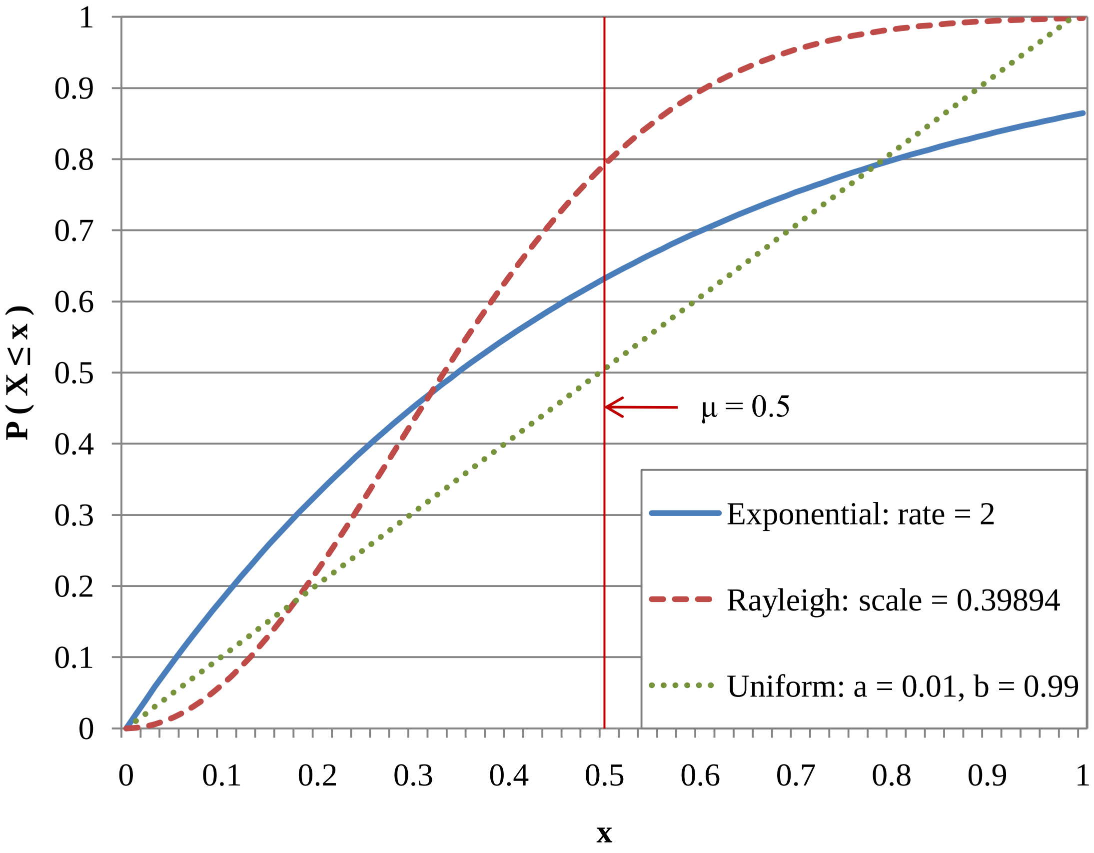

Figure 7: Cumulative distribution functions of the transmission time distribution

The last factor having a significant effect on the metric is the number of nodes in the network, accounting for 7.62% of the total variation. Although a larger network equates to more nodes creating queries and agents, it also provides each node with additional neighbors to transmit them to, thus reducing their probability of receiving a packet. As such, the 5000-node network had a lower mean rate of arrivals per node (μ = 0.223, σ = 0.038) than the 500-node network (μ = 0.249, σ = 0.044).

The next three factors, in order of greatest effect, are TTL, agent expiration distribution and query expiration distribution, equating to approximately 3% of the total variation.

4.4.2. Proportion of Query Failures

The residual plots for the proportion of query failures ANOVA model are shown in Figure 8. The residuals in the normal probability plot are linear, with the exception of a few outliers causing a slight s-curve appearance in the tail. The histogram also shows the residuals following a normal distribution curve, thus the normal distribution of the residuals is verified. The outliers causing the slight s-curve are listed in Appendix C. Of

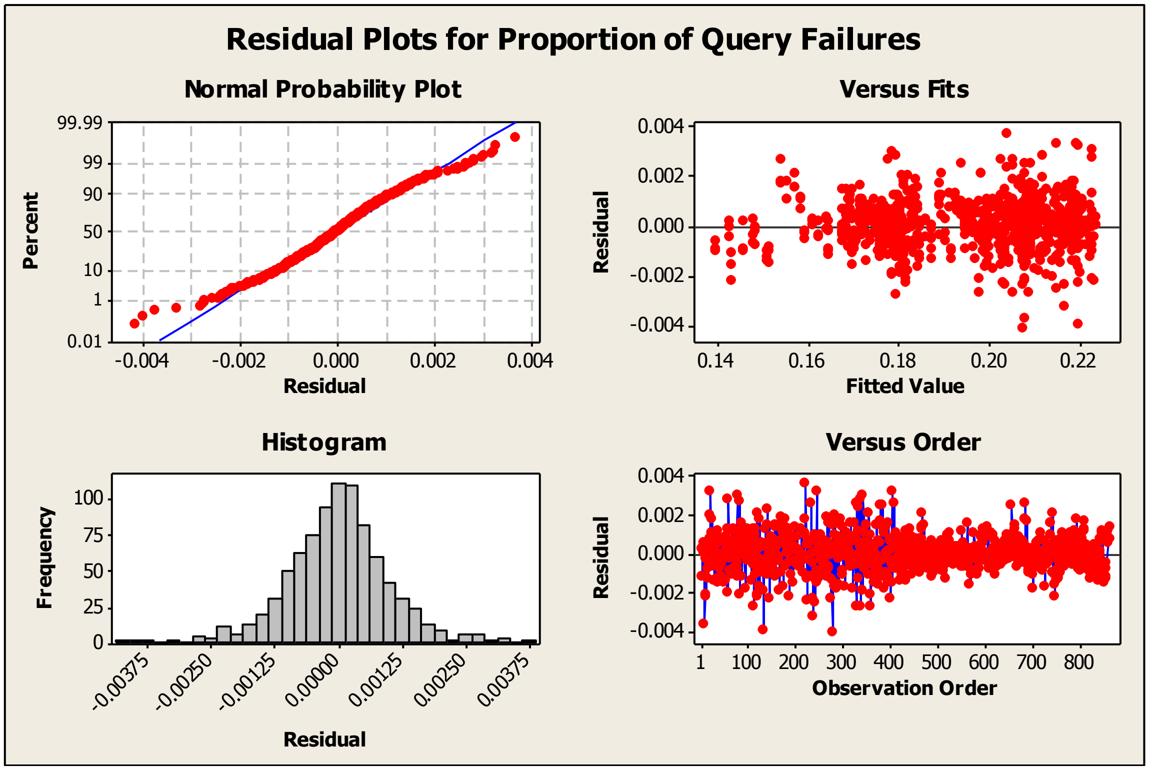

Figure 8: ANOVA residual plots for the proportion of query failures

the 27 outliers, whose residual values are greater and less than 0.002, all but two have a factor of N = 500. There are no other trends indicated involving the other factors. As stated earlier, it is expected less error will exist with a larger node population.

The residual versus fits plot show the residuals growing slightly larger as the fit increases. This normally indicates the variance is not constant or a data transform is necessary; however neither a logarithmic or square root transform improves the results. The scale of the residuals is so small, however, that the largest residual error is only -0.0041585. With such a small scale taken into account, it is assumed the residuals are fairly randomly scattered, verifying the data's constant variance.

The ANOVA table for the proportion of query failures is found in Appendix B. In analyzing the sum of squares for each factor and interaction of factors, four factors were found to account for approximately 93% of the total variation on the proportion of query failures. These factors are listed in the abbreviated Table 6. The factor with the greatest effect was the transmission time distribution, accounting for 73.99% of the observed variation. Query failures, as defined earlier, occur in a node's transmission queue. It is understandable, then, that this factor has a large effect on the proportion of queries that fail. The Rayleigh transmission time distribution with a scale of 0.39894 resulted in the lowest proportion of query failures ($\mu = 0.173$, $\sigma = 0.011$), compared to the exponential distribution with a rate of 2 ($\mu = 0.201$, $\sigma = 0.009$) and uniform distribution with a minimum time of 0.01 seconds and maximum time of 0.99 seconds ($\mu = 0.212$, $\sigma = 0.010$). Again, as shown in Figure 7, the Rayleigh distribution provides a greater probability of a lower transmission time, which gives queries less time to expire in the transmission queue, thus the lower probability of query failures.

Table 6: Factors with the main effect on the proportion of query failures

Source	*DF*	*SS*	*MS*	*F*	*P*
Transmission Time Dist	2	0.2357684	0.1178842	114773.59	0.000
Query Expiration Dist	1	0.0280699	0.0280699	27329.20	0.000
N	1	0.0195575	0.0195575	19041.43	0.000
Agent Interarrival Dist	1	0.0138451	0.0138451	13479.81	0.000
…	…	…	…	…	…
Total	863	0.3186607			

The factor with the next greatest effect is the query expiration distribution, accounting for approximately 8.81% of the total variance. Trials with a uniform query expiration distribution with a minimum lifetime of 0.01 seconds and maximum lifetime of 3.99 seconds on average had a lower proportion of query failures ($\mu = 0.189$, $\sigma = 0.020$) than the exponential distribution with a rate of 0.5 ($\mu = 0.201$, $\sigma = 0.017$). As shown in Figure 9, the probability of the exponential query expiration distribution being less than the mean of two seconds is 63.2%, compared to 50.1% for the uniform distribution. As a result, the uniform distribution generally provides queries with a longer lifetime, thus a lower proportion of query failures, than the exponential distribution.

The number of nodes in the network (6.14%) and the agent interarrival distribution (4.34%) had the next greatest effects on the total variance. The network with 5000 nodes, on average, had a lower proportion of query failures ($\mu = 0.190$, $\sigma = 0.020$) than the network with 500 nodes ($\mu = 0.200$, $\sigma = 0.017$). In a network with a greater number of nodes, more agents will propagate through the network, resulting in a greater amount of informed nodes. Likewise, a much greater number of queries will be circulating the

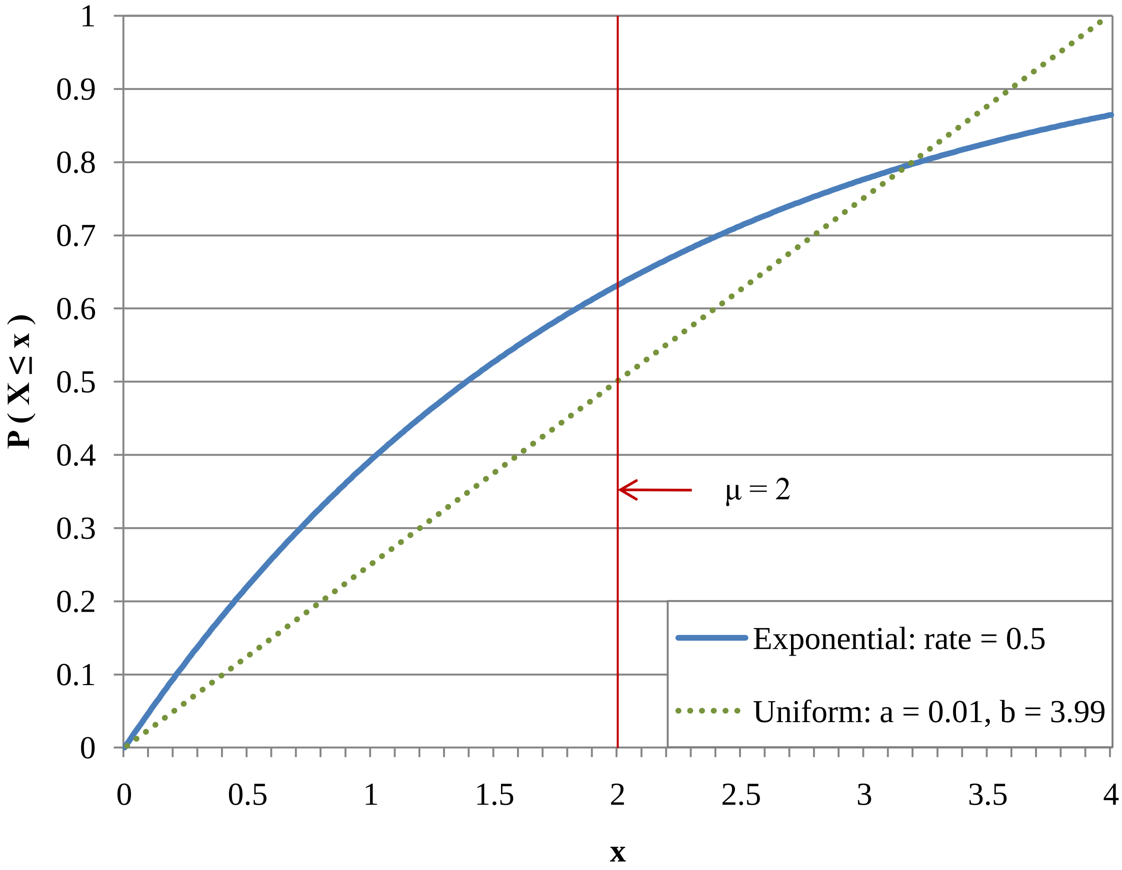

Figure 9: Cumulative distribution functions of the query expiration distribution

network, many of which will discover the same informed nodes, resulting in a lower probability of failure than a 500 node model.

The exponential agent interarrival distribution with a rate of 0.01 produced a lower proportion of query failures ($\mu = 0.191$, $\sigma = 0.020$) than the exponential distribution with a rate of 0.005 ($\mu = 0.200$, $\sigma = 0.018$). With a rate of 0.01, agents are generated an average of once every 100 seconds, compared to once every 200 seconds if the rate is 0.005. Thus, the rate of 0.005 has a higher probability of query failures, as fewer agents are propagating through the network, resulting in fewer informed nodes.

The next factors, in order of greatest effect, are the interaction between the number of nodes and the query expiration distribution (1.94%), the query interarrival distribution (0.86%), TTL (0.73%), and the interaction between the number of nodes and the agent interarrival distribution (0.68%). All other factors and interactions did not have a significant effect on the total variation of the proportion of query failures.

4.4.3. Mean Proportion of Time a Node is Uninformed

The residual plots for the mean proportion of time a node is uninformed are shown in Figure 10. The residuals in the normal probability plot are linear, and the histogram shows the residuals follow a normal distribution curve, thus the normal distribution of the residuals is verified. No visual trends are detected within the residual versus fits and

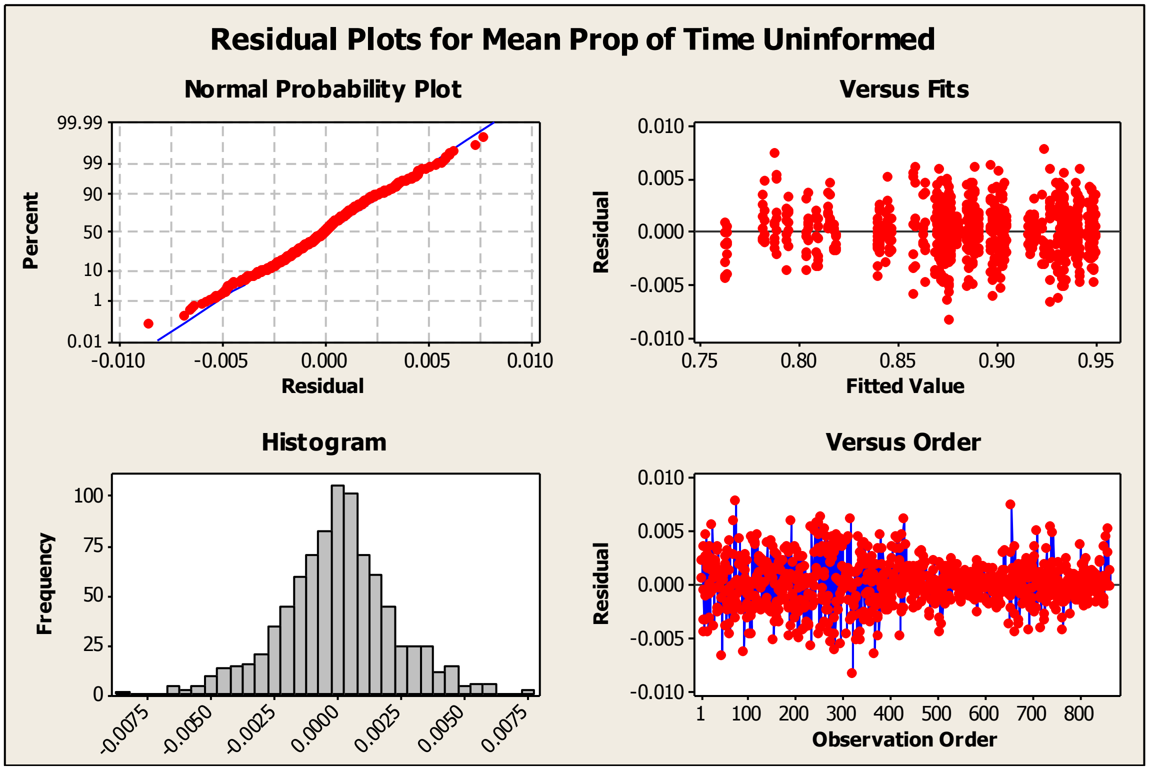

Figure 10: ANOVA residual plots for the mean proportion of time a node is uninformed

residual versus order plots, thus the independence and constant variation are verified.

The sum of squares of the mean proportion of time uninformed ANOVA table indicated that approximately 86% of the total variation was determined by three factors: the agent interarrival distribution, number of nodes in the network, and the TTL of an agent, as shown in the abbreviated Table 7. The agent interarrival distribution accounted for 54.85% of the total variation. In general, an exponential agent interarrival distribution with a rate of 0.01 resulted in a lower proportion of time a node was uninformed (μ = 0.865, SD = 0.031) than an exponential distribution with a rate of 0.005 (μ = 0.926, SD = 0.020). The exponential distribution with a rate of 0.01 generates agents twice as fast as the distribution with a rate of 0.005, resulting in a greater number of informed nodes and a lower proportion of time a node is uninformed.

The number of nodes in the network accounted for approximately 17.96% of the variation. A network with 5000 nodes had a lower proportion of time a node was uninformed (μ = 0.881, SD = 0.044) than a network with 500 nodes (μ = 0.912, SD = 0.030). In a larger network, many more agents are generated and transmitted through the network. This results in a greater number of informed nodes, thus a lower proportion of time uninformed.

Table 7: Factors with the main effect on the mean proportion of time a node is uninformed

Source	*DF*	*SS*	*MS*	*F*	*P*
Agent Interarrival Dist	1	0.977680	0.977680	192452.39	0.000
N	1	0.320114	0.320114	63013.19	0.000
TTL	2	0.235472	0.117746	23175.85	0.000
…	…	…	…	…	…
Total	863	1.782563			

The number of hops an agent could make, or TTL, accounted for approximately 13.21% of the total variance. A TTL of 25, on average, resulted in lower mean proportion of time uninformed ($\mu = 0.879$, SD = 0.050), compared to a TTL of 15 ($\mu = 0.883$, SD = 0.046) and 5 ($\mu = 0.916$, SD = 0.029). It is assumed an agent's TTL will expire before its expiration lifetime, thus with a greater TTL, an agent is able to inform many more nodes, resulting in a lower proportion of time nodes are uninformed.

The remaining effects are primarily accounted for by the agent expiration distribution (3%), transmission time distribution (1.4%), and interactions between the number of nodes and the agent TTL (2.5%), number of nodes and the agent interarrival distribution (1.9%), agent expiration distribution and the agent TTL (1.3%), agent interarrival distribution and the agent TTL (1.1%), and the number of nodes and the agent expiration distribution (0.72%). Unlike the other metrics, in which each factor had at least some effect upon the performance, the query expiration distribution and all of its interactions with other factors had no effect upon the proportion of time a node was uninformed. This is logical, as a node is only informed by agents and remains informed until its lifetime expires. Queries discovering an informed node have no effect upon its informed status. The other interactions did not have a significant effect on the mean proportion of time a node is uninformed.

4.5. Summary

This chapter presents the design of the OPNET node model and its components, as well as describes the modified rumor routing protocol and the user-controlled parameters. It discusses the metrics used to measure network performance and explains how they are calculated. The updated OPNET code is shown to perform, with 95% confidence, nearly identically to the original code [MBK+08].

The simulation trials are described, and results presented. The query interarrival distribution parameter has the greatest effect upon a network’s energy consumption, accounting for 73.64% of the total variation. An exponential query interarrival distribution with a rate of 0.4 queries/second/node uses 25.78% less power than an exponential distribution with a rate of 0.6 queries/second/node. The transmission time distribution accounts for 73.99% of the total variation of the proportion of query failures. Of three distributions, each with a mean of 0.5 seconds, the proportion of query failures using a Rayleigh transmission time distribution is 14.23% less than an exponential distribution and 18.46% less than a uniform distribution. Finally, the agent interarrival distribution has the greatest effect upon the mean proportion of time a node is uninformed, accounting for 54.85% of the total variation. The mean proportion of time a node is uninformed using an exponential agent interarrival distribution with a rate of 0.005 is 6.59% higher than an exponential distribution with a rate of 0.01.

5. Conclusions and Recommendations

The effects packet-related parameters have upon the performance of a modified rumor routing protocol using various distributions within a large-scale wireless sensor network are determined by modeling the protocol and WSN within OPNET, a discrete-time simulator.

5.1. Results

The following results are determined from the simulation:

5.1.1. Mean Rate of Packet Arrivals per Node

The query interarrival distribution has the greatest effect (73.64%) upon the total variation in the mean rate of packet arrivals per node. This is due to queries being generated at a rate approximately 4-5 times greater than agents. Furthermore, an exponential query interarrival distribution with a rate of 0.4 queries/second/node ($\mu = 0.200$, $\sigma = 0.019$) uses 25.78% less power than an exponential distribution with a rate of 0.6 queries/second/node ($\mu = 0.270$, $\sigma = 0.020$). Thus, to prolong the life of a WSN and its nodes, the user should be primarily concerned with minimizing the rate at which queries are generated.

Other factors with a large effect on the total variation are the transmission time distribution (11.11%) and the number of nodes in the network (7.62%).

5.1.2. Proportion of Query Failures

The total variation in the proportion of query failures is most affected (73.99%) by the transmission time distribution. Since query failures occur while awaiting transmission in the transmission queue, increased transmission time will increase query failures. Of three distributions, each with a mean value of 0.5 seconds, the proportion of query failures using a Rayleigh distribution ($\mu = 0.173$, $\sigma = 0.011$) is 14.23% less than an exponential distribution ($\mu = 0.201$, $\sigma = 0.009$) and 18.46% less than a uniform distribution ($\mu = 0.212$, $\sigma = 0.010$). Thus, to achieve a lower proportion of query failures, the user should minimize the nodes' transmission time and use a Rayleigh transmission time distribution.

Other factors with a significant effect on the total variation in proportion of query failures are the query expiration distribution (8.81%), number of nodes in the network (6.14%), and the agent interarrival distribution (4.34%).

5.1.3. Mean Proportion of Time a Node is Uninformed

Factors affecting the total variation of the mean proportion of time a node is uninformed are more diverse than the previous two metrics. Still, the agent interarrival distribution has the greatest effect (54.85%). Using an exponential distribution with a rate of 0.005 ($\mu = 0.926$, SD = 0.020), the mean proportion of time a node is uninformed is 6.59% higher than an exponential distribution with a rate of 0.01 ($\mu = 0.865$, SD = 0.031). Agents are needed to inform nodes, thus to reduce the proportion of time a node is uninformed, the user should maximize the rate at which agents are generated.

Other factors with a large effect on the mean proportion of time a node is uninformed are the number of nodes in the network (17.96%) and TTL (13.21%). The query expiration distribution, and all its interactions, has no effect on this metric.

5.2. Contributions

This research demonstrated that certain factors have a greater effect upon the performance of a large-scale, wireless sensor network using a rumor routing protocol with limited packet lifetimes. It also showed that varying the distribution of certain functions, while maintaining the same mean value, affects network performance. Enhancing the simulation model to measure the mean proportion of time a node is uninformed will support the development of future analytic models.

5.3. Future Research

There are several areas in which additional research could be performed. These include:

- Analyze the effect of applying various distributions to the agent and query interarrival distributions. In this research, they used only exponential distributions with varying rates.
- Apply other distributions to the protocol parameters and examine their effect on the network performance.
- Develop an analytic model to account for different distributions, and use the simulation model to verify the results.

- Examine each factor’s effect with various network topologies and/or obstructions.
- Integrate node mobility into the simulation model and analyze the effect it has upon each factor.

Appendix A: List of Acronyms

ACK: Acknowledgement

AEA: Adaptive Election Algorithm

ANOVA: Analysis of Variance

B-MAC: Berkeley Media Access Control

CCA: Clear Channel Assessment

CSMA: Carrier Sense Multiple Access

CTS: Clear to Send

ECN: Explicit Contention Notification

FRTS: Future Request to Send

HCL: High Contention Level

LCL: Low Contention Level

LPL: Low Power Listening

MAC: Medium Access Control

MORRP: Modified Rumor Routing Protocol

MPTNU: Mean Proportion of Time a Node is Uninformed

MRPAN: Mean Rate of Packet Arrivals per Node

NP: Neighbor Protocol

PD-MAC: Packets Decision Medium Access Control

PQF: Proportion of Query Failures

RTS: Request To Send

S-MAC: Sensor Medium Access Control

SEP:	Schedule Exchange Protocol
SLR:	Straight Line Routing
T-MAC:	Timeout Medium Access Control
TDMA:	Time Division Multiple Access
TRAMA:	Traffic-Adaptive Medium Access
TSBQ:	Trajectory-based Selective Broadcast Query
TTL:	Time to Live
WSN:	Wireless Sensor Network
Z-MAC:	Zebra Medium Access Control
ZRR:	Zonal Rumor Routing

Appendix B: ANOVA Tables

Results for: Mean Rate of Arrivals per Node

```
Factor                   Type   Levels
N                        fixed       2
Agent Interarrival Dist  fixed       2
Agent Expiration Dist    fixed       2
Query Interarrival Dist  fixed       2
Query Expiration Dist    fixed       2
Transmission Time Dist   fixed       3
TTL                      fixed       3

Factor                   Values
N                         500, 5000
Agent Interarrival Dist  exponential; rate = 0.005, exponential; rate = 0.01
Agent Expiration Dist    exponential; rate = 0.3, uniform; a = 0, b = 6.67
Query Interarrival Dist  exponential; rate = 0.04, exponential; rate = 0.06
Query Expiration Dist    exponential; rate = 0.5, uniform; a = 0.01, b = 3.99
Transmission Time Dist   exponential; rate = 2, rayleigh; scale = 0.39894,
uniform; a = 0.01, b = 0.99
TTL                       5, 15, 25


Analysis of Variance for Mean Rate of Arrivals per Node

Source                          DF        SS        MS           F      P
N                                1  0.121273  0.121273    48089.45  0.000
Agent Interarrival Dist          1  0.005952  0.005952     2360.34  0.000
Agent Expiration Dist            1  0.011803  0.011803     4680.37  0.000
Query Interarrival Dist          1  1.171213  1.171213   464431.36  0.000
Query Expiration Dist            1  0.011715  0.011715     4645.37  0.000
Transmission Time Dist           2  0.176692  0.088346    35032.58  0.000
TTL                              2  0.032054  0.016027     6355.32  0.000
N*Agent Interarrival Dist        1  0.007655  0.007655     3035.70  0.000
N*Agent Expiration Dist          1  0.002072  0.002072      821.69  0.000
N*Query Interarrival Dist        1  0.003808  0.003808     1510.09  0.000
N*Query Expiration Dist          1  0.001757  0.001757      696.73  0.000
N*Transmission Time Dist         2  0.009151  0.004576     1814.40  0.000
N*TTL                            2  0.009054  0.004527     1795.12  0.000
Agent Interarrival Dist*         1  0.000681  0.000681      269.92  0.000
  Agent Expiration Dist
Agent Interarrival Dist*         1  0.004910  0.004910     1947.11  0.000
  Query Interarrival Dist
Agent Interarrival Dist*         1  0.000263  0.000263      104.23  0.000
  Query Expiration Dist
Agent Interarrival Dist*         2  0.000611  0.000305      121.12  0.000
  Transmission Time Dist
Agent Interarrival Dist*TTL      2  0.006440  0.003220     1276.86  0.000
Agent Expiration Dist*           1  0.000128  0.000128       50.74  0.000
  Query Interarrival Dist
Agent Expiration Dist*           2  0.000181  0.000090       35.81  0.000
  Transmission Time Dist
Agent Expiration Dist*TTL        2  0.001736  0.000868      344.24  0.000
Query Interarrival Dist*         1  0.000349  0.000349      138.55  0.000
  Query Expiration Dist
Query Interarrival Dist*         2  0.004099  0.002049      812.66  0.000
```

```
  Transmission Time Dist
Query Interarrival Dist*TTL      2  0.000888  0.000444     175.97  0.000
Query Expiration Dist*           2  0.000362  0.000181      71.84  0.000
  Transmission Time Dist
Transmission Time Dist*TTL       4  0.000253  0.000063      25.10  0.000
N*Agent Interarrival Dist*       1  0.000282  0.000282     111.85  0.000
  Query Interarrival Dist
N*Agent Interarrival Dist*       2  0.000237  0.000118      46.94  0.000
  Transmission Time Dist
N*Agent Interarrival Dist*TTL    2  0.000143  0.000071      28.28  0.000
N*Agent Expiration Dist*TTL      2  0.000862  0.000431     170.87  0.000
N*Query Interarrival Dist*       2  0.000286  0.000143      56.66  0.000
  Transmission Time Dist
N*Query Interarrival Dist*TTL    2  0.000370  0.000185      73.42  0.000
N*Transmission Time Dist*TTL     4  0.000650  0.000163      64.49  0.000
Agent Interarrival Dist*         2  0.000114  0.000057      22.65  0.000
  Agent Expiration Dist*TTL
Agent Interarrival Dist*         2  0.000102  0.000051      20.32  0.000
  Query Interarrival Dist*
  Transmission Time Dist
Agent Interarrival Dist*         4  0.000202  0.000050      19.99  0.000
  Transmission Time Dist*TTL
Agent Expiration Dist*           4  0.000203  0.000051      20.08  0.000
  Transmission Time Dist*TTL
Error                          796  0.002007  0.000003
Total                          863  1.590559

S = 0.00158802   R-Sq = 99.87%   R-Sq(adj) = 99.86%
```

Results for: Proportion of Query Failures

```
Factor                   Type   Levels
N                        fixed       2
Agent Interarrival Dist  fixed       2
Agent Expiration Dist    fixed       2
Query Interarrival Dist  fixed       2
Query Expiration Dist    fixed       2
Transmission Time Dist   fixed       3
TTL                      fixed       3

Factor                   Values
N                         500, 5000
Agent Interarrival Dist  exponential; rate = 0.005, exponential; rate = 0.01
Agent Expiration Dist    exponential; rate = 0.3, uniform; a = 0, b = 6.67
Query Interarrival Dist  exponential; rate = 0.04, exponential; rate = 0.06
Query Expiration Dist    exponential; rate = 0.5, uniform; a = 0.01, b = 3.99
Transmission Time Dist   exponential; rate = 2, rayleigh; scale = 0.39894,
uniform; a = 0.01, b = 0.99
TTL                       5, 15, 25


Analysis of Variance for Proportion of Query Failures

Source                          DF         SS         MS          F       P
N                                1  0.0195575  0.0195575   19041.43   0.000
Agent Interarrival Dist          1  0.0138451  0.0138451   13479.81   0.000
Agent Expiration Dist            1  0.0006659  0.0006659     648.36   0.000
Query Interarrival Dist          1  0.0027301  0.0027301    2658.02   0.000
Query Expiration Dist            1  0.0280699  0.0280699   27329.20   0.000
Transmission Time Dist           2  0.2357684  0.1178842  114773.59   0.000
TTL                              2  0.0023421  0.0011710    1140.13   0.000
N*Agent Interarrival Dist        1  0.0021804  0.0021804    2122.85   0.000
N*Agent Expiration Dist          1  0.0003719  0.0003719     362.05   0.000
N*Query Expiration Dist          1  0.0061914  0.0061914    6027.99   0.000
N*Transmission Time Dist         2  0.0001027  0.0000513      49.99   0.000
N*TTL                            2  0.0011312  0.0005656     550.69   0.000
Agent Interarrival Dist*         1  0.0000793  0.0000793      77.17   0.000
  Agent Expiration Dist
Agent Interarrival Dist*         1  0.0021254  0.0021254    2069.36   0.000
  Query Expiration Dist
Agent Interarrival Dist*         2  0.0000246  0.0000123      11.97   0.000
  Transmission Time Dist
Agent Interarrival Dist*TTL      2  0.0002465  0.0001233     120.01   0.000
Agent Expiration Dist*           1  0.0000296  0.0000296      28.86   0.000
  Query Expiration Dist
Agent Expiration Dist*TTL        2  0.0003308  0.0001654     161.04   0.000
Query Interarrival Dist*         1  0.0000103  0.0000103      10.03   0.002
  Query Expiration Dist
Query Interarrival Dist*         2  0.0000560  0.0000280      27.27   0.000
  Transmission Time Dist
Query Expiration Dist*           2  0.0002711  0.0001356     131.99   0.000
  Transmission Time Dist
Query Expiration Dist*TTL        2  0.0003298  0.0001649     160.54   0.000
N*Agent Interarrival Dist*       1  0.0000485  0.0000485      47.22   0.000
  Agent Expiration Dist
N*Agent Interarrival Dist*       1  0.0005047  0.0005047     491.37   0.000
  Query Expiration Dist
N*Agent Interarrival Dist*       2  0.0000159  0.0000080       7.74   0.000
  Transmission Time Dist
```

```
N*Agent Interarrival Dist*TTL    2  0.0001125  0.0000562      54.74  0.000
N*Agent Expiration Dist*         1  0.0000743  0.0000743      72.34  0.000
  Query Expiration Dist
N*Agent Expiration Dist*TTL      2  0.0001640  0.0000820      79.84  0.000
N*Query Expiration Dist*         2  0.0001629  0.0000814      79.28  0.000
  Transmission Time Dist
N*Query Expiration Dist*TTL      2  0.0001952  0.0000976      95.03  0.000
Agent Interarrival Dist*         2  0.0000275  0.0000137      13.36  0.000
  Agent Expiration Dist*TTL
Agent Interarrival Dist*         2  0.0000408  0.0000204      19.87  0.000
  Query Expiration Dist*
  Transmission Time Dist
Agent Expiration Dist*           2  0.0000204  0.0000102       9.94  0.000
  Query Expiration Dist*TTL
Error                          812  0.0008340  0.0000010
Total                          863  0.3186607

S = 0.00101346   R-Sq = 99.74%   R-Sq(adj) = 99.72%
```

Results for: Mean Proportion of Time Uninformed per Node

```
Factor                   Type   Levels
N                        fixed       2
Agent Interarrival Dist  fixed       2
Agent Expiration Dist    fixed       2
Query Interarrival Dist  fixed       2
Transmission Time Dist   fixed       3
TTL                      fixed       3

Factor                   Values
N                         500, 5000
Agent Interarrival Dist  exponential; rate = 0.005, exponential; rate = 0.01
Agent Expiration Dist    exponential; rate = 0.3, uniform; a = 0, b = 6.67
Query Interarrival Dist  exponential; rate = 0.04, exponential; rate = 0.06
Transmission Time Dist   exponential; rate = 2, rayleigh; scale = 0.39894,
uniform; a = 0.01, b = 0.99
TTL                       5, 15, 25


Analysis of Variance for Mean Prop of Time Uninformed

Source                        DF        SS        MS          F      P
N                              1  0.320114  0.320114   63013.19  0.000
Agent Interarrival Dist        1  0.977680  0.977680  192452.39  0.000
Agent Expiration Dist          1  0.054379  0.054379   10704.35  0.000
Query Interarrival Dist        1  0.000197  0.000197      38.86  0.000
Transmission Time Dist         2  0.024561  0.012280    2417.33  0.000
TTL                            2  0.235472  0.117736   23175.85  0.000
N*Agent Interarrival Dist      1  0.033837  0.033837    6660.69  0.000
N*Agent Expiration Dist        1  0.012859  0.012859    2531.17  0.000
N*Transmission Time Dist       2  0.003209  0.001604     315.81  0.000
N*TTL                          2  0.044775  0.022388    4406.89  0.000
Agent Interarrival Dist*       1  0.004714  0.004714     927.91  0.000
  Agent Expiration Dist
Agent Interarrival Dist*       2  0.001811  0.000905     178.23  0.000
  Transmission Time Dist
Agent Interarrival Dist*TTL    2  0.020067  0.010034    1975.10  0.000
Agent Expiration Dist*TTL      2  0.023957  0.011979    2357.93  0.000
Transmission Time Dist*TTL     4  0.005378  0.001345     264.66  0.000
N*Agent Interarrival Dist*     1  0.001073  0.001073     211.20  0.000
  Agent Expiration Dist
N*Agent Interarrival Dist*     2  0.000366  0.000183      36.06  0.000
  Transmission Time Dist
N*Agent Interarrival Dist*TTL  2  0.004041  0.002020     397.68  0.000
N*Agent Expiration Dist*TTL    2  0.005969  0.002984     587.45  0.000
N*Transmission Time Dist*TTL   4  0.001069  0.000267      52.61  0.000
Agent Interarrival Dist*       2  0.001921  0.000960     189.03  0.000
  Agent Expiration Dist*TTL
Agent Interarrival Dist*       4  0.000303  0.000076      14.89  0.000
  Transmission Time Dist*TTL
N*Agent Interarrival Dist*     2  0.000528  0.000264      51.92  0.000
  Agent Expiration Dist*TTL
N*Agent Interarrival Dist*     4  0.000145  0.000036       7.12  0.000
  Transmission Time Dist*TTL
Error                        815  0.004140  0.000005
Total                        863  1.782563

S = 0.00225391   R-Sq = 99.77%   R-Sq(adj) = 99.75%
```

Appendix C: Outliers from Residual Plots

Mean Rate of Arrivals per Node

Residual	N	Agent Int. Dist.	Agent Exp. Dist.	Query Int. Dist.	Query Exp. Dist.	Transmission Time Dist.	TTL
0.0055696	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
0.0055272	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	5
0.0050063	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	25
0.0048865	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	15
0.0047916	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
0.0041444	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	exponential; rate = 2	15
0.0040432	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	5
0.0038593	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	exponential; rate = 2	25
0.0036696	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	15
-0.0037378	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	25
-0.0037987	5000	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
-0.0038725	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	15
-0.0038760	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	5
-0.0038761	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	25
-0.0039014	5000	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
-0.0039102	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	5
-0.0040859	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	exponential; rate = 2	25
-0.0041429	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	15
-0.0042150	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	exponential; rate = 2	15
-0.0043253	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
-0.0044340	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	15
-0.0046740	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	15
-0.0047642	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
-0.0051492	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	5

Proportion of Query Failures

Residual	N	Agent Int. Dist.	Agent Exp. Dist.	Query Int. Dist.	Query Exp. Dist.	Transmission Time Dist.	TTL
0.0036611	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	25
0.0032419	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	5
0.0032394	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
0.0031923	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	15
0.0030258	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	5
0.0029860	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
0.0028289	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	5
0.0028057	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	5
0.0027487	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
0.0026382	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	5
0.0026372	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	15
0.0026360	5000	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	5
0.0026146	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	exponential; rate = 2	5
0.0024905	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	25
0.0024823	5000	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	25
0.0024747	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	15
0.0023926	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	15
0.0023002	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	exponential; rate = 2	5
-0.0025554	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
-0.0027299	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	5
-0.0027324	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	5
-0.0027429	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	exponential; rate = 2	15
-0.0027902	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	5
-0.0032961	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	5
-0.0037160	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	15
-0.0039733	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	15
-0.0041585	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	15

Mean Proportion of Time a Node is Uninformed

Residual	N	Agent Int. Dist.	Agent Exp. Dist.	Query Int. Dist.	Query Exp. Dist.	Transmission Time Dist.	TTL
0.0076601	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
0.0073073	5000	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	25
0.0062006	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	5
0.0059873	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	15
0.0059851	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
0.0058619	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
0.0058567	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	15
0.0057271	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	25
0.0056904	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	exponential; rate = 2	5
0.0054086	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	15
0.0053633	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
0.0053268	5000	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	exponential; rate = 2	25
0.0050923	5000	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	15
0.0050904	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
-0.0049986	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	15
-0.0050017	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
-0.0052010	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
-0.0052892	500	exponential; rate = 0.005	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
-0.0052967	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	exponential; rate = 2	15
-0.0053309	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	15
-0.0055121	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	exponential; rate = 2	5
-0.0057565	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	uniform; a = 0.01, b = 0.99	15
-0.0059297	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.04	exponential; rate = 0.5	rayleigh; scale = 0.39894	25
-0.0062782	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	exponential; rate = 0.5	rayleigh; scale = 0.39894	5
-0.0064134	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	exponential; rate = 2	25
-0.0065433	500	exponential; rate = 0.01	uniform; a = 0, b = 6.67	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	25
-0.0067478	500	exponential; rate = 0.005	exponential; rate = 0.3	exponential; rate = 0.04	uniform; a = 0.01, b = 3.99	rayleigh; scale = 0.39894	15
-0.0085121	500	exponential; rate = 0.01	exponential; rate = 0.3	exponential; rate = 0.06	uniform; a = 0.01, b = 3.99	uniform; a = 0.01, b = 0.99	15

Bibliography

[BE02] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 22-31, 2002.

[BTJ05] T. Banka, G. Tandon and A. Jayasumana, "Zonal rumor routing for wireless sensor networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, vol. II, pp. 562-567, 2005.

[CSC05] C. F. Chou, J. J. Su and C. Y. Chen, "Straight line routing for wireless sensor networks," in *Proceedings of the 10th IEEE Symposium on Computers and Communications*, pp. 110-115, 2005.

[EETA07] "Miniaturized wireless sensor node from IMEC Research Center." http://www.eetasia.com/ARTICLES/2007NOV/C/EEOL_2007NOV08_STECH_INTD_NP_fig10.jpg. 08 November 2007

[JWZ+08] P. Ji, C. Wu, Y. Zhang and Z. Jia, "Research of an energy-aware MAC protocol in wireless sensor network," in *Chinese Control and Decision Conference*, pp. 4686-4690, 2008.

[L04] F. L. Lewis, "Wireless Sensor Networks," in *Smart Environments: Technologies, Protocols, and Applications*, 2004.

[LAN97] LAN MAN Standards Committee of the IEEE Computer Society, *Wireless LAN medium access control (MAC) and physical layer (PHY) specification*, IEEE Std. 802.11, 1997.

[MBK+07] C. R. Mann, R. O. Baldwin, J. P. Kharoufeh and B. E. Mullins, "A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks," in *Telecommunication Systems*, pp. 67-86, 2007.

[MBK+08] C. R. Mann, R. O. Baldwin, J. P. Kharoufeh and B. E. Mullins, "Energy-efficient search for finite-lifetime resources in sensor networks with time-constrained queries" in *ACM SIGMOBILE Mobile Computing and Communications Review*, pp. 31-39, 2008.

[PHC04] J. Polastre, J. Hill, D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, pp. 95-107, 2004.

[RWM+06] I. Rhee, A.Warrier, J.Min and L. Xu, “DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks,” in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 190-201, 2006.

[ROG06] V. Rajendran, K. Obraczka and J.J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” in *Wireless Networks*, vol. 12, pp. 63-78, 2006.

[RWA+08] I. Rhee, A. Warrier, M. Aia, J. Min and M.L. Sichitiu, “Z-MAC: a hybrid MAC for wireless sensor networks,” in *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511-524, 2008.

[VL03] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 171-180, 2003.

[YHE02] W. Ye, J. Heidemann and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1567-1576, 2002.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)* 25-03-2010	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* Sep 2008 – Mar 2010

4. TITLE AND SUBTITLE

Analysis of a Rumor Routing Protocol with Limited Packet Lifetimes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Peter R. Francik, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/10-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

INTENTIONALLY LEFT BLANK

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Wireless sensor networks require specialized protocols that conserve power and minimize network traffic. Therefore, it is vitally important to analyze how the parameters of a protocol affect these metrics. In doing so, a more efficient protocol can be developed.

This research evaluates how the number of nodes in a network, time between generated agents, lifetime of agents, number of agent transmissions, time between generated queries, lifetime of queries, and node transmission time affect a modified rumor routing protocol for a large-scale, wireless sensor network. Furthermore, it analyzes how the probability distribution of certain protocol parameters affects the network performance.

The time between generated queries had the greatest effect upon a network's energy consumption, accounting for 73.64% of the total variation. An exponential query interarrival distribution with a rate of 0.4 queries/second/node used 25.78% less power than an exponential distribution with a rate of 0.6 queries/second/node. The node transmission time was liable for 73.99% of the total variation in proportion of query failures. Of three distributions, each with a mean of 0.5 seconds, the proportion of query failures using a Rayleigh transmission time distribution was 14.23% less than an exponential distribution and 18.46% less than a uniform distribution. Lastly, 54.85% of the total variation in the mean proportion of time a node is uninformed was a result of the time between generated agents. The mean proportion of time a node is uninformed using an exponential agent interarrival distribution with a rate of 0.005 was 6.59% higher than an exponential distribution with a rate of 0.01.

15. SUBJECT TERMS

Wireless sensor network, search protocol, energy efficiency, reliability, optimization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Rusty O. Baldwin, PhD
U	U	U	UU	85	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4445 rusty.baldwin@afit.edu

Standard Form 298 (Rev: 8-8)
Prescribed by ANSI Std. Z39-18