7-25-2012

STER: Sensor-Triggered Efficient Routing

Kristopher Langston

Recommended Citation

Langston, Kristopher, "STER: Sensor-Triggered Efficient Routing" (2012). *Electronic Theses and Dissertations*. 528.
https://digitalcommons.memphis.edu/etd/528

STER: SENSOR-TRIGGERED EFFICIENT ROUTING

by

Kristopher Langston

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Computer Engineering

The University of Memphis

August, 2012

Abstract

Langston, Kristopher Patrick. M.S. The University of Memphis. August 2012. STER: Sensor-Triggered Efficient Routing. Major Professor: Dr. Aaron Robinson

Wireless sensor networks (WSN) have become powerful tools for gathering and monitoring environmental data. These networking systems can be utilized for many different applications due to their autonomy, ability to withstand harsh conditions, and the reduced cost associated with their collection of data. These characteristics are beneficial across a wide range of applications including those specific to the military, environmental, industrial, and medical industries. Additionally, they become increasingly more relevant in remote sensing applications where size weight and power trade-offs are of particular importance. Conversely, these applications also demonstrate the Achilles heel of a large percentage of WSNs in that they run on limited power sources. Thus, energy efficiency is a major concern and therefore a significant amount of research has been dedicated to identifying methods of making WSNs as energy efficient as possible. The purpose of this paper is to detail a reactive wireless sensor network protocol that will minimize network overhead and energy consumption in an effort to provide longevity to the overall network. The underlying components of the Sensor-Triggered Efficient Routing protocol, STER, are covered and the asynchronous handshaking method used to transmit data between the sending and receiving nodes is also described. The power consumption performance results of STER are compared to those obtained from other protocols in the current literature. The data shows that implementation of the STER protocol expends the least amount of energy compared to the other wireless sensor network protocols designed for energy efficiency. Based on results, it can be concluded that specific applications where a spatially dense nodal network is feasible will have an increased life span with the implementation of STER.

Table of Contents

List of Tables

I. INTRODUCTION

A. *Wireless Sensor Networks*

Wireless sensor networks (WSN) have become useful for gathering, processing, and transmitting data. There has been a niche for the usefulness of WSNs due to a number of factors including, equipment affordability, the range of operating conditions, and the different kinds of information that the system can gather. WSNs so far have been implemented in environmental, industrial, healthcare, and home automation industries.

Generally, WSNs consist of a base station and a network of nodes. Each node contains a radio transceiver antenna, a microcontroller, and an array of different sensors. Each node also has its own battery power source [6]. The nodes in the network can send and receive data from other nodes and the base station. Nodes are also able transmit data using either a multi-hop technique where information propagates from node to node until it reaches the destination or, in some cases, a single hop in which data travels directly from the source node to destination

Given the wide range of potential WSN applications and the numerous advantages inherent in their inclusion, the most significant drawback to their use must be addressed. Because the nodes are autonomous and typically run off of batteries, each node has only a finite supply of energy. Since the amount of power is limited, the requirement for the collection of nodes to be as energy efficient as possible, while still performing as a network, is of primary importance. For example, assume that a critical node happened to deplete all of its power. We define a critical node as one that is necessary to propagate the data to the base station for processing. If this node loses power, the network, or branch of a network, is deemed ineffective, even if the other nodes are functional. Thus, research continues to focus on creating new methodologies that result in more energy efficient networks via software protocols and various techniques.

B. Previous Research

WSN protocols establish the guidelines for data transfer between nodes from one location to the next. Generally protocols determine many aspects of communication and can include characteristics of the system such as error handling, authentication, and signaling. Wireless sensing node communication can generally be described using the communication architecture for its protocol. The communication architecture can be designed to target different layers in the protocol stack [11]. Specifically, there are five layers in the stack termed the application layer, transport layer, network layer, physical layer, and the data link layer. Each has a hand in how information gets passed along and each provides different opportunities to achieve energy optimization. This section provides the details of the current research whose focus targets one of these layers to improve energy. Every protocol that has been created will fall into one of these layers.

All protocols can be either reactive, proactive, or hybrid. A reactive protocol is one where the nodes only send data when there is pertinent data to transmit from the source to the sink node. A proactive protocol does the opposite of a reactive protocol. Instead of waiting for some significant environmental trigger to occur that is worthy of reporting, proactive networks consistently send data about the environment at regular intervals. While they are both great approaches, there is a tradeoff between resolution and energy efficiency. The reactive protocols are more energy efficient because of the significant decrease in radio use over time, but they do so at the cost of environmental resolution. Proactive networks provide better temporal resolution because it captures and propagates data at regular intervals, but at the cost of energy consumption. A hybrid protocol tries to optimize the benefits and minimize the weaknesses proactive and reactive protocols. Every protocol will fall into either one of these types as well.

Many protocols attempt to conserve energy by restricting the number of “working” nodes to the minimum amount required to carry the workload. The other nodes not involved in the communication process remain in a “sleep” state, where a node in said state will typically deactivate the device components, including the radio and sensors. Through scheduling, nodes cycle between sleeping and working. The idea is that with each node doing less work over time, the lifetime of the overall network will improve. The Probing Environment and Adaptive Sleeping (PEAS) protocol uses this approach by establishing a dynamic sleep schedule for the nodes in the network [12]. It performs this task by using an algorithm that focuses on three main aspects. The first aspect determines which nodes should be working. The second controls how a node waking up from a sleep state decides whether or not to go back to sleep. The final aspect involves dynamically determining the average time that sleeping nodes should stay sleep before it is time to wake up again. The protocol performs this by initially having all nodes in a sleep state with the wakeup times for each exponentially distributed across the nodes. When a node wakes up, it sends a PROBE message to the neighboring nodes. The node will begin working if no REPLY is received. Otherwise it goes back to sleep for another random time. Another protocol mechanism addresses energy consumption by keeping a certain amount of nodes awake to maintain overall network connectivity. The number of nodes is determined by the settings broadcasted by the base station to each node in the network. Simulation results indicated that the network failed when about 38% of the nodes failed and although that proved that a protocol could be robust and adaptive in an environment, the protocol's performance and efficiency can be improved.

The Balancing Energy-Aware Sensing Management (BESM) protocol takes what PEAS has done and builds from it. This protocol uses the residual energy level of each individual node to determine which nodes should sleep longer and thus conserve their energy [13]. The protocol consists of five states for each node: initialization, sleeping, probing, active, and dead. In the initialization phase, each node goes through the time synching, node localization, topology formation, etc. The sleeping phase involves all nodes going into sleep mode for a random time. The next phase is probing, where a node wakes up and sends out a PROBE message and waits to hear a response. If no response is received, the node begins working. The nodes continue to work until the nodes that are sleeping wake up at their designated time. At this time they will send out the PROBE packet with that node's energy. If there are any nodes whose energy consumption is lower than the current threshold, that node will REPLY with a negative vote and then go to sleep for a random time. If the probing node doesn't hear a REPLY or doesn't receive a negative response then it will begin working and become an active node. This continues until the nodes consume all of their power. This method improves on the PEAS protocol and simulates 3 different scenarios with the amount of nodes increasing from 100, to 200, to 400 respectively. BESM outperforms PEAS in both coverage area lifetime and data delivery.

While the BESM and PEAS protocols conserve energy by allowing a node subset to sleep while other nodes handle the workload, another common approach to minimize workload of nodes over time. Diffusion and clustering are representative of this methodology. Through diffusion, information is transferred from one location to the other by choosing the best path at the time. This usually involves some kind of network “energy awareness”. The best path commonly goes towards the path with either the least amount of energy used or the path whose nodes have the most energy to spare overall. Clustering handles data propagation differently, but with the same goal in mind. The idea is that if a node transmits smaller data to a closer receiver, then the radio transmission would consume less energy. Over time that particular node, and subsequently the network overall, will survive longer. There have been different protocols using these two methodologies, and some examples of these methodologies will be touched on later in the paper.

The Energy Aware Protocol is a reactive protocol that implements the directed diffusion concept and builds on the principle. The protocol’s main idea is to increase the network survivability by choosing the best path for data to propagate from source to sink. The “best” path is chosen as the one with the highest residual energy. The goal of the protocol is to provide a more graceful degradation of the network over time. The protocol sets up by a sink node broadcasting an interest message. The receivers will then send an interest message to a subset of nodes. This process continues until a predetermined subset of source nodes is reached. The source nodes will then gather exploratory data to find the lowest-latency path back to the sink. The data passed in the exploratory data from the source to the sink includes the rolling distance from the source and the corresponding residual energy. The residual energy is overall energy of the data path from source to sink. It is calculated as the rolling sum of each node’s individual energy. When data needs to be transmitted from the source to the sink, the nodes refer to a stored setup table of residual energy totals for each path. Thus, the path containing the highest residual energy value is chosen as the best path.

The Two-Tier Data Dissemination (TTDD) protocol, like the Energy Aware protocol, is a protocol that uses the dissemination methodology. However, instead of catering towards energy optimization, TTDD aims to more effectively handle data propagation efficiently over a large-scale, mobile network. Unlike the Energy Aware protocol, TTDD is a proactive protocol that constructs a grid structure between the nodes so that the source nodes can receive data while multiple sink nodes are on the move. TTDD uses two separate tiers to determine how data is propagated. When a sink node has data that needs to be transmitted, the first-tier states that the data be transmitted to the local grid point, while the second tier states that the grid points aggregate all of the data from multiple sink nodes and forward it to other grid points until the data reaches the source. The grid is determined by the source based off of the location of the node relative to the area of the network and the predetermined size of the grid cells. The grid points, also referred to as dissemination points, are determined such that the distance from the center of the grid is $\alpha/2$, where α is the area of the grid cell. During simulation, TTDD was compared to a sink-oriented data dissemination protocol like Directed Diffusion. TTDD performs similarly when it comes to the success rate of data being transmitted. Both perform in the 90th percentile when there are up to 4 sources, but drops to about 80% when the source count increases to 8. The energy consumption also performed similarly. The time delay from sink to source, however, increases with TTDD slightly less than direct diffusion as the number of sinks increases.

Low Energy Adaptive Clustering Hierarchy (LEACH) is a protocol that utilizes the clustering framework for data propagation. LEACH is a proactive protocol. This kind of protocol allows the network to take periodic samples of data consistent basis. LEACH utilizes a clustering network topology where subsets of nodes are composed of member nodes and one node is designated as the cluster head [14][4]. The cluster members collect data and send it to the cluster head. The job of the cluster head is to aggregate the data and transmit that data to the base station. Each cluster uses a Time Division Multiple Access (TDMA) schedule so that each node has a turn to send data to the cluster head without conflict. Additionally, the cluster head creates and manages the schedule. The network of cluster heads use Code Division Multiple Access (CDMA) to propagate the data to the source or base station. The protocol goes through rounds where a cluster head is chosen independently by determining how long it has been since each node was last a cluster head. The heads will then advertise with the neighboring nodes that it has become the cluster head, and the member nodes decides whose cluster to join by the head with the strongest advertisement signal. Once all the clusters are composed, the cluster head will create a TDMA table and broadcast the table to all the cluster members so that each node will know when it is their turn to send data to the head. Data propagation then moves on to the next phase. The next phase is the steady-state phase where the data transmission occurs. Based on the TDMA table, the cluster head receives the transmitted data from the members of the cluster. After all of the data is collected, the cluster head then transmits the aggregated data on to the base station for analysis. The nodes benefit from having their radio off until the TDMA dictates that it’s their turn to send information to the cluster head because it reduces energy consumption. Even though this method makes a decent attempt to optimize energy consumption, it has drawbacks. One drawback is that the head nodes energy dissipation is unbalanced over time. Consequently, some nodes die significantly faster over other nodes. One cause of this issue is due to the random selection process of the protocol. Another drawback is that the protocol cannot cover a large area. Some research has gone into building on this protocol to address some of the issues with LEACH.

There are other protocols that build off of LEACH, with each one addressing one or more of its stated issues. TL-LEACH is a protocol that builds on the LEACH by using a two-tier hierarchy to manage data propagation. During the setup phase, there is a random selection of not only the cluster head, but a secondary head as well. The cluster head only communicates with the secondary head. The secondary aggregates the data and passes it on to the cluster head for data forwarding to the base station. By dividing up the task of the cluster head, TL-LEACH further lowers the amount of energy that a node uses over time.

Other protocols built on LEACH are E-LEACH, TL-LEACH, M-LEACH, LEACH-C, and V-LEACH. Energy LEACH (E-LEACH) focuses on improving the selection process of the cluster head. Instead of choosing the cluster head through random selection, nodes that have higher residual energy will have a higher probability of being the cluster head. TL-LEACH improves on the data forwarding by cluster heads using the multi-hop method for forwarding using the neighboring cluster heads instead of the cluster head transmitting directly to the base station. This requires less power and thus clusters far away die less frequently. M-LEACH is similar to TL- LEACH in that it uses the multi-hop design for data forwarding using the cluster head, finding the most optimal path to get from the source to the base station. LEACH-C is a protocol that focuses on better cluster formation by using residual energy and GPS location to determine the best clusters. V-LEACH is a protocol that takes into account the possibility of a cluster head dying while being the cluster head during that round. The basic concept is that in case this situation occurs there would be a backup node that will act as the vice cluster head. If the cluster head dies, then the vice cluster head will become the new cluster head. The vice cluster head is selected during the selection process. Generally, all of these protocols that looked to improve upon LEACH improved the duration of the network's survivability.

Although one of the major concerns for the protocol is optimizing the energy utility of a network, a network must also be dependable. Furthermore, in certain applications, dependability can be a higher priority. Real-time networks have been the way to go when it comes to having a Quality of Service (QoS) standard. A lot of research focus has gone into balancing the trade-off between improving energy efficiency and minimizing data transmission delay. Protocols such as SPEED, REDRP, and RPTAW, along with others are just some of the examples of research that focuses on providing real-time data while trying to be energy-efficient as well, and will touch on some of them.

Real Time Power Aware Framework (RPTAW) is a real-time communication system that is designed to be energy efficient by using the clustering concept as well [2]. A real-time network works differently from a proactive network. Instead of periodically sending data to the base station for analysis, it only sends relevant data based on the system. The cluster roles are slightly different from LEACH, adding an additional role referred to as a Relay Node. A cluster would consist of a Relay Node, a Cluster Head, and multiple Cluster Members. Cluster members play the same role as the cluster members in LEACH. The cluster head's role, however, in LEACH has been broken up in RPTAW. The cluster head aggregates the data from the cluster member node, while the cluster head periodically transmits data to the relay node. The relay node is tasked with transmitting data from cluster to cluster or from cluster to sink. The network goes through 3 phases: the initiation phase, the election phase, and the data transfer phase. The initiation phase determines the structure of the cellular grid that is best for the network. It is assumed that the nodes are aware of their location and that the area is arranged in a uniformly dense area. The grid is divided into a number of cluster cells. Each cell covers a small area within the network. The election phase is where the roles of the nodes are determined. The cluster head is selected based on the node that has the strongest residual energy and appoints the relay node based on the node that has the next highest residual energy of that cluster. The final phase is the data transfer phase. Here the cluster head broadcasts a beacon packet to the cluster to help synchronize the cluster. The beacon helps create a TDMA system with time slots allotted for each node to transmit data to the cluster head. This also allows the cluster head to have a slot in which it sends all the aggregated data to the relay node for inter-cluster transmitting. If a cluster member is not transmitting, it can go to sleep to reserve power. However, it has to wake up in time to receive the next beacon broadcasting. After cluster head sends the data, the cluster goes back into election phase to determine the new cluster head. The relay node has an associated energy threshold and is the only node that stays constant. Once that relay node drops below its power threshold value, it notifies the cluster head. The cluster head then assigns another node to be the relay node based on the highest residual energy at that time.

SPEED is another real-time protocol that is designed to handle localized wireless communication and optimize data transmission. It functions by attempting to find the most efficient data path. This is done through feedback control design with stateless algorithm methods. Each node in the SPEED network has a Beacon Exchange system to periodically broadcast its location to the neighboring nodes. The Stateless Non-deterministic Geographic Forwarding (SGNF) algorithm handles the data forwarding and decides which node route is the shortest path based on current location of the nodes [1]. Note that node location is time dependent since nodes in this protocol can be mobile or stationary. The algorithm uses a combination of calculated relay speeds with relation to the desired speeds and the load of the nodes to determine the best route. The nodes with a combination of higher speeds and lower workload have a higher probability of being selected using an exponential distribution. SGNF design serves two functions, the first being the ability to pick the best forwarding node to choose that meets the real-time constraints. The second function is that it helps balance the load over the network. The Neighborhood Feedback Loop (NFL) also has a hand in maintaining end-to-end delay by maintaining a Relay Ratio. The ratio is used for determining and dynamically updating the relay requirement that helps meet the real-time constraint. The NFL keeps track of the percentage of failed packet hops and tries to converge the percentage of misses down to as close to 0 as possible. A failed packet is defined as a packet that has either missed its relay deadline or was not received due to some conflict (i.e. collision or dying node). Simulation results show that the protocol was able to effectively find the best route and handle congestion to reduce to end-to-end packet delay. However, the miss ratio required to achieve that success was about 20%. When observing the energy consumption, SPEED performed about the same as the other protocols tested in the simulation. Yet when congestion occurs, the energy consumption rises as compared to those same protocols.

Reactive Energy Decisive Routing Protocol (REDRP) is a real-time protocol that takes residual energy of the nodes in the network to determine which route to take [3]. Each node will have knowledge of the neighboring nodes' residual energy. The initiation phase of the protocol has the sink broadcasting a distance packet with each node getting a different value from the sink indicating how far it is from the sink. The nodes then go into sensing mode using minimal amount of energy until some event occurs. The event can either be an environmental stimulation worth reporting or a packet to establish a routing path. If a node senses relevant data, the Route Discovery phase is initiated to determine a route. It will then send a special packet to its neighbors requesting a route to deliver data to the sink. These special packets are passed forward to neighboring nodes and consist of Packet_C, Hop_Count, Prev_ID, Source_ID, and RP_ID data. The Packet_C is the packet ID. Hop_Count represents the number of current hops and is modified as each packet is forwarded. Prev_ID, Source_ID, and RP_ID represent the previous node, the current node, and the route ID respectively. If a node receives the packet and the Hop_Count is greater than the number of nodes, then that packet is dropped. The receiving node also sends a Reply_Packet to the sender with the Node_ID, the residual energy, and the distance value of that node. Once a node receives the Reply packet the original node will determine which node to choose for its data route by selecting the node with the smaller distance followed by the node with the highest energy. Once that node is chosen, another request packet is sent to the chosen node asking that node to find the next node in the routing path. This continues until the node reaches the sink. The sink then sends a Confirm packet back down that path, ending at the source. Once the path is verified with the Confirm packet, the nodes transmit the data along the path while using the RP_ID to keep the paths and the data transmission unique. Once the transmission is complete, the nodes go back into sensing mode.

In special cases, an interruption in the route created by a node can occur due to a node on the route path sensing some action before it had a chance to forward the original transmission. When this occurs, the node sensing the secondary event will notify the previous node on the route with a RP_Adj, which includes the Node_ID. The previous node will then remove that node off of the list, choose the next available node along the list stored from the Route Discovery phase and will notify that node as well as the interrupted node. The new node will begin the route discovery path to the sink as the interrupted node begins its own discovery phase. Once the data gets to the new node, it will weigh the option of whether to return to the original path or use the new distance and residual energy based path created in its own route discovery. Simulation results reflected that the energy dissipation over time was slower when compared to other protocols like LEACH and PEGASIS.

PEGASIS is a protocol that builds off of the same concept as LEACH. PEGASIS attempts to make a chain with the nodes so that the nodes communicate with close neighbors. The idea is that these nodes take turns transmitting data to the base station in a manner that reduces the average energy consumed over time. By creating this chain of nodes, the nodes in turn would use less energy overtime since the nodes would only have to use enough transmission power necessary to pass data on to its closest neighbor. When the chain is linked, data is passed using a greedy algorithm method. The node passing the data on the chain to the base station is selected randomly. If a node in the chain dies, then the link bypasses that dead node to the remaining nodes in the chain. Simulation results show that the PEGASIS outperforms the LEACH protocol between 100 – 300%, with the performance improving as the number of nodes in the system dies [16].

The one thing that all of these protocols have in common is that they all use the radio to communicate with neighboring nodes or the base station. Another common aspect is that they all attempt to be more energy efficient by limiting the use of the radio. My theory aims to drive down the use of the radio significantly unless a node is about to send and receive data. The protocol developed in this thesis will communicate using onboard sensor triggering to establish communications. Therefore, the use of idle radio transmission and overhead schedules to transmit data from the field to the base station is effectively bypassed.

C. Motivation

With a fairly new way of using existing technology, there are generally going to be more problems to solve than originally expected. Energy consumption has become such a significant issue, it takes precedence in what many system designs attempt to optimize. This overshadows more common networking concerns, like fault tolerance, transmission speeds, and other QoS performance guidelines found in traditional network systems. Due to the priority of energy concerns, research focus has concentrated more on how to lower power consumption using different topologies and location awareness techniques. Effective as these methods are, there are only so many unique network topology variations available to attempt to maximize energy efficiency.

So taking look at other alternatives for energy minimization is critical. The idea focuses on the actual power used by network nodes. There are three major operations that draw power on a node: operating power, radio activity, and sensors activity. The biggest draw of power used to operate a device on the node is the radio. Additionally, the one common observation that every method included in this treatment exhibited is that they all had to have at least one node in the network with its radio on to receive data from neighboring nodes. If there isn't any potentially worthwhile data being forwarded, then the radio just idles. Thus, energy is expended on the process of data reception and transmission with little to show for it. If this is the case, then the energy could have been reserved and potentially used to sustain the operating power of the node. So the current proposition is stated as follows: is it possible to somehow cut down the time the radio is actually on?

Consider the radio's primary use on a network node. For all intents and purposes, its principal use is to transmit information back and forth throughout the network. With that said, if there is no need to transfer data, then the radio is serving no real purpose outside of air sniffing. So if the idling can be averted, then perhaps valuable energy resources can be maintained. However, this solution also has an inherent problem as well. Stated succinctly, if the radio isn't monitoring the air for data how can the network communicate between nodes to provide an indication that the radio needs to be powered on and prepared for regular network traffic. This thesis addressed this issue as follows. Since every node has a set of sensors, these sensors can theoretically be used to trigger neighboring nodes. In this case, it is proposed that that sensor's output be used as a triggering event to state that a node is ready to send data and to prepare to receive it. The overriding idea is that the protocol can be flexible enough to implement a system using any available set of sensors, any possible communication triggering medium(such as an acoustic or optical), and not rely too heavily on specific hardware requirements to operate effectively. It is proposed that the STEP algorithm will accomplish all of these criteria while minimizing the use of the radio and essentially reducing energy consumption.

D. Application

WSNs have a number of features that designers find attractive when searching for solutions to data transmission applications. Versatility is the first of these features. WSN versatility results in the ability to integrate a network of wireless sensors into virtually any existing technology. The versatility is due to the ability for a network system to be application-specific. Not all protocol designs can be incorporated into all applications. However, select protocol designs coupled with select hardware platforms could better suit application requirements. Durability is the second desirable feature. WSNs are durable enough to be assimilated into various environmental conditions. WSN's durability can be credited to the inherent properties of the hardware and radio transmission. The hardware provides the ability to function in harsh climates. Another layer of durability can be credited to protocol design as well. Some protocols are able to perform effectively even after certain nodes in the network die. These two characteristics of durability and versatility gives WSNs increased opportunities for implementation in an ever expanding list of situations and industries. Therefore, it is of no surprise that the use of Sensor-

Triggered Efficient Routing (STER) may be limited to only a subset of these industries and scenarios.

With the current available technology, STER is limited in the amount of applications that can utilize its potential. Additionally, STER will be limited by the resolution and sensitivity of the underlying sensors used to initiate the “triggering” effect. For example, a limitation of STER is that it must have a “line of sight” to enable a dependable triggering medium. This requirement also has secondary consequences such as its effect on how dense the nodes must be in the network in order for the protocol to be effective. Applications that best fit STER’s capabilities are generally those related to level sensitive environmental monitoring where a spatially dense network can be deployed. Some examples of these applications can include identification of environmental anomalies such as sudden temperature changes, changes in air pressure or vibration in the proximate area surrounding the network. STER would also be beneficial to a wide variety of military applications including systems that detect intrusion of potential hostiles in an open or closed environment. Other applications can be implemented across various industries, including applications such as a local area alarm system for a room in a house, recording earthquake tremors, theft prevention system, etc. Generally speaking, any form of area system that monitors a determined static space is a good condition to implement STER.

II. ENERGY CONSUMPTION

With devices and systems pushing to become more wireless and mobile at a rapid pace, energy efficiency and the lifetime before human intervention becomes a concern that all mobile communication devices must face. So much focus has been placed on energy efficiency that every piece of hardware component is designed with energy conservation in mind. This is coupled with software design capabilities that allow for control over the amount of power needed for certain processes. For example, current wireless nodes offer software control over radio transmission power and processor speed. Unfortunately, it isn’t possible to discuss energy consumption without referring to hardware because the circuit design determines the amount of current draw and the amount of voltage load needed to perform certain actions. Thus, for the sake of establishing a basis for the energy models considered in this thesis, the MicaZ has been selected as the platform under test. The MicaZ platform is equipped with the CC240 2.4 GHz radio transmitter. Note that although the energy values may change with different hardware designs, the amount of energy efficiency for each protocol should remain relatively consistent across different platforms.

The MicaZ is equipped with the Atmega128L that allows for different power levels and processor states. This prevents the node from wasting unnecessary power when needed. According to Krämer, the power saving capabilities of the Atmega chip significantly reduces the power consumption of the processor [10]. Table 1 below shows that the processor can reduce the amount of energy used based on the operation mode [9].

Table 1
Atmega128L Operation Mode

Operation	Avg. Current (mA)
Busy (mul)	8.65
Busy (jmp)	8.73
No Op	7.69
Idle	3.88
ADC	1.32
Standby	0.25
Save	0.14

The experiments were run under the condition of the power supply was set to 3.1 V and all other components were turned off. Also, according to the MicaZ datasheet, the active and sleep mode draws about 8 mA and 15 µA respectively [7]. A processor in active mode is a processor that is running a process or a set of processes. When a processor is in sleep mode, the processor minimizes energy because it suspends the process that it is running and only draws enough current to remain powered on. Sleep mode is different from a processor that is turned off because it doesn’t have to reboot and is able to save the current state that the system is in. A processor that is turned off would have to reboot and would lose the information stored in RAM and would potentially have to start from the beginning of the program or process. Sleep mode is preferred because it prevents a full reset of the system while still being able to minimize energy consumption.

The CC2420 transceiver is another component that has power-saving capabilities. The transceiver has multiple operation modes with different energy consumption levels. The following table based on the MicaZ datasheet shows the amount of current that is consumed from the different transceiver states [7].

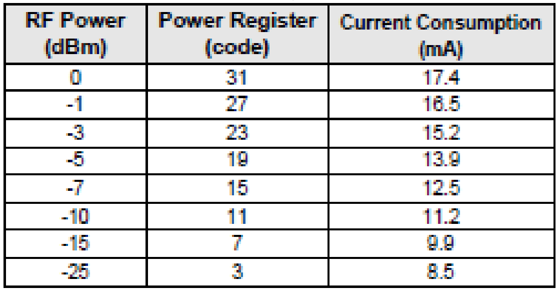
Table 2
Current Draw of CC2420 Operation Modes

Operation	Avg. Current (mA)
Receive(Rx)	19.7
Transmit(Tx) 0 db	17.4
Transmit(Tx) -5 db	14
Transmit(Tx) -10 db	11
Idle	0.02
Sleep	0.001

The receive operation consumes the most power out of the transceiver, followed by the transmit operation, idle, and sleep respectively. Some protocols balance the burden of being hit by the such a high load by creating schedules where a subset of nodes are in receive mode while the others subsets are in transmit mode, idle, or sleep. A scheduling system such as this attempt to take advantage of the significant difference in energy consumption the states have, particularly with the Tx/Rx and the idle or sleep mode.

The transceivers ability to control and transmit at certain power levels also provides another avenue for energy reduction. Higher power levels require more energy consumption. Table 3 shows the transmission power with relation to the energy consumption. It also shows a little more resolution in regards to how much the energy consumption changes based on the power level of the transmission. According to the MEMSIC’s Mote Processor Radio & Mote Interface Boards User Manual, using lower power also increases the likelihood of data being dropped and interference from occurring.

Table 3
CC2420 Output power and Typical Current Consumption



RF Power (dBm)	Power Register (code)	Current Consumption (mA)
0	31	17.4
-1	27	16.5
-3	23	15.2
-5	19	13.9
-7	15	12.5
-10	11	11.2
-15	7	9.9
-25	3	8.5

The receive operation doesn’t have such a luxury. However, TinyOS does provide an alternative to help conserve energy while receiving. An operation called Low Power Listening (LPL) allows an energy saving technique so that the 17.4 mA doesn’t take such a heavy toll on nodes in receive mode. LPL provides a duty cycling scheme such that the node cycles the radio between the on and off position and performs a receive check in the process while on. This helps relieve the heavy burden of that mode.

Since this protocol isn’t hardware specific, the other components will vary. Other components may include a light sensor, acoustic sensor, temperature sensor, magnetometer, accelerometer, etc. These components generally have minimal impact on the protocol’s energy consumption scheme, thus the energy consumption of sensors can be deemed negligible.

As stated previously, the task of receiving transmissions induces the biggest energy burden on a mote. LPL helps alleviates the burden by cycling the radio between active and inactive states. Implementing LPL consumes about 10 times less energy than listening for the full designated time period [17]. With the transceiver drawing about 17 mA of current, implementing the LPL method can cut down the reception task to about 1.7 mA. Sensor triggering is able to rival the energy performance. The amount of energy needed to generate a buzzer event using the sounder is about 1.2 mA [18]. The energy needed for the microphone to listen for the desired tone is about 500 µA [19].

There are many models that measures performance for a protocol. Unfortunately, many of the models were only designed as a performance metric for a particular protocol. Therefore, a model based on the hardware of WSNs must be considered as a barometer for energy efficiency across different protocols. There are few power consumption models that speak in general to where protocols can be modeled by them. There is no way to determine what the power consumption of a network is without depending on the hardware of the network or making assumptions about the energy levels of certain states in the protocol. Since the sensing components are being deemed negligible, the main focus of the nodes will be on the energy consumption of radio activity.

Since most models are based on power, not current, the power would have to be calculated. Using the simple power formula, the amount of power a node generates in a single radio state can be determined. The power formula is as follows;

$$P = V * I \tag{1}$$

where P equates the power, V is voltage, and I represents the current. Assuming the voltage is constant at 3.3 V, the data from Table 2 is used to determine the energy. Table 4 reflects the power associated with transceiver.

Table 4
Power Consumption of CC2420 Operation Modes

Operation	Avg. Power (mW)
Receive(Rx)	65.01
Transmit(Tx) 0 db	57.42
Transmit(Tx) -5 db	46.2
Transmit(Tx) -10 db	36.3
Idle	0.066
Sleep	0.0033

Once the power is calculated, the energy is given by the following equation:

$$E = P * t \tag{2}$$

Where t is time in seconds. Assuming a 1 Mbps transmission rate, the time to send or receive 1 bit is then

$$1\ bit = \frac{1}{1\ Mbps} = 1\ \mu sec \tag{3}$$

Table 5 holds the consumption data corresponding to the different transmission states.

Table 5
Energy Consumption of CC2420 Operation Modes

Operation	Avg. Energy (µJ/bit)
Receive(Rx)	0.06501
Transmit(Tx) 0 db	0.05742
Transmit(Tx) -5 db	0.0462
Transmit(Tx) -10 db	0.0363
Idle	0.000066
Sleep	0.0000033

Basic Power Consumption Model: The basic power consumption model according to Hemptsted creates a simple equation based off of the following diagram [8].

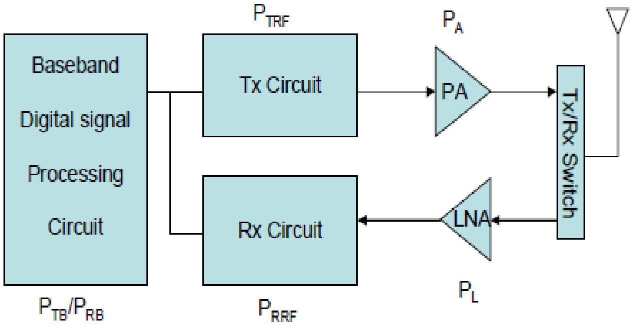


Fig.1. A typical communication Model Structure on a typical WSN device.

Fig. 1 illustrates what a typical communication system would be composed of with a typical WSN device. P_{TB}/P_{RB} reflects the power consumption based on the transmission and reception for the DSP circuit. P_{TRF}/P_{RRF} represents the power consumption attributed to the sending and receiving circuit at the front-end of the communication module. P_A and P_L denotes the power consumption of the power amplifier and low-noise amplifier respectively. Based on Fig. 1, the simple power formula for transmitting and receiving respectively is:

$$P_T(d) = P_{TB} + P_{TRF} + P_A(d) = P_{T0} + P_A(d) \tag{4}$$

$$P_R = P_{RB} + P_{RRF} + P_L = P_{R0} \tag{5}$$

Since the DSP and front-end circuitry is going to be a constant, $P_{TB}$and P_{TRF} can be reduced to constant P_{T0}. Similarly, P_{RB} and P_{RRF} are reduced to constant P_{R0}.

The amount of power required to transmit data via radio communication depends on the distance required to transmit the data from source to destination. According to Hempsted's use of a simple class A power amplifier circuit, the total power consumption of the power amplifier directly correlates to the DC power input, P_{DC}, and the ratio of the RF output power to DC input power is known as the drain efficiency [8][5]. The following equation is noted as:

$$\eta = \frac{P_{Tx}}{P_{DC}} \tag{6}$$

Since P_A can simply be represented by the amount of input power input, (4) can be modified to:

$$P_T(d) = P_{T0} + P_{Tx}(d)/\eta \tag{7}$$

where $P_{Tx}$is the power that is sent to the antenna using the power amplifier.

The channel model takes communication path loss into consideration when receiving data. The following equation models the power expended during reception considering those obstacles as:

$$P_{Rx} = P_{Tx}/(A * d^{\alpha}) \tag{8}$$

Where P_{Rx} represents the power used for the receiving node when using its transmitter to receive transmissions from the air. The $P_{Tx}$is the aforementioned power generated from the transmitting node and   is the distance traveled.   is determined the by characteristics of the antenna. The variable   represents the freespace and is generally the constant 2. The freespace's value can increase, however, based on the number of obstacles. Combining (6) and (7) gives the basic Power Consumption Model which is:

$$P_T(d) = P_{T0} + \left(\frac{P_{Rx} * A * d^{\alpha}}{\eta}\right) \tag{9}$$

In order to have the most reliable reception for a receiving node, it's best to use the minimum required power to achieve the best Signal-to-Interference and Noise-Ratio. Replacing P_{Rx} with the minimal receiving power, P_{Rx-min} can slightly modify (9) to become:

$$P_T(d) = P_{T0} + (\frac{\varepsilon * d^{\alpha}}{\eta}) \tag{10}$$

where ε is the constant $P_{Rx} * A$. For the receiving node:

$$P_R = P_{R0} \tag{11}$$

Multi-Hop Power Consumption Model: The basic power consumption model is utilized to determine the power consumption of a node at a local level. However, a network consists of a collection of nodes and thus calculation of network power dissipation has to consider all nodes and their corresponding power consumption methods to determine the overall use of energy. In order to determine the amount of power that is consumed over a network, a model of a single-hop or multi-hop network must be developed. Beginning with the channel model and assuming only path loss, then the simplest model would be a 1-D network topology model. A graphical model of these topologies is shown in Fig.2 [8].

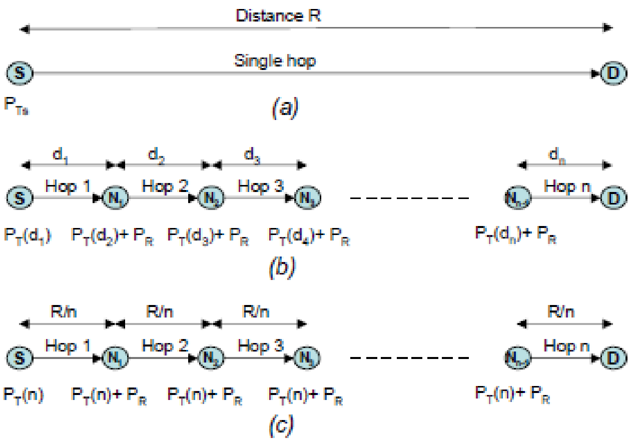


Fig. 2. (a) A single hop model, (b) a multi-hop network with random distances between each node, (c) a multi-hop network with equidistant nodes

Each hop model consists of a source node and a destination node, or sink node, and is separated by some distance . According to Heinzelman, using a single-hop method traveling far distances kills off a network faster than using multiple hops with a fraction of the distance [15].

Multi-hop networks, on the other hand, will have a source node, a destination node, and a set of intermediate nodes positioned between the source and destination node to relay the data for a multi-hop network. $P_T(d)$ denotes the amount of power used for sending a transmission some distance, d. $P_R$is the power used for receiving a transmission. The destination node's power consumption is ignored because of the assumption that it is connected to some external power supply, therefore the energy it expends is not important in this setup.

Fig. 2b and Fig. 2c are slightly different in a subtle, but significant way. Fig. 2b depicts the scenario where the nodes are placed arbitrarily such that the distances between each node are equal. Therefore distance d_i and d_j are not guaranteed to be the same value. Under this scenario and given knowledge of the parameters of the channel model equation, the power of the overall network can be determined as:

$$P(n) = (n-1)P_{R0} + nP_{T0} + \frac{\varepsilon}{\eta}\sum_{i=1}^{n} d_i^{\alpha} \tag{12}$$

where i is the number of current hops, ranging from 1 to n. Fig. 2c takes into consideration that all the nodes along the data path from the source node to the destination node are equidistant from each other. Because of this, the distance variable can be replaced by total distance R over total hops n . The network power consumption equation will slightly differ from (12) and will be:

$$P(n) = (n-1)P_{R0} + nP_{T0} + \frac{n * \varepsilon * (\frac{R}{n})^{\alpha}}{\eta} \tag{13}$$

III. METHODOLOGY

STER has been designed such that performance is maintained while the overhead is reduced to provide efficient energy consumption. The protocol uses a simple system that allows data to propagate towards the base station without sacrificing one or more motes' energy reserve. Also with the lack of a scheduling system to keep motes updated, the network can spend more time collecting data instead of having to stop data collection to reassign roles. STER uses a concept similar to the REDRP protocol with its initiation phase. Thus, nodes further away from the base station are assigned higher distance values. Data propagates by comparing the distance value at the current node to the value of the node receiving the data and only accepting the data that is valid. STER builds on this routine by tying the Node ID to the distance packet during the initiation phase. This allows the node to be assimilated back into the network seamlessly without requiring the rest of the nodes in the network to be affected. Since the Node ID is programmed into the node, it can seamlessly go back to contributing to the network after loss of power or functionality. The node would simply have to request the distance value from the base station. The base station would have a table of distance values for all of the nodes and will be tied to the Node ID. STER should also be versatile enough such that it would not be limited to just using one particular sensor triggering device. If an application calls for a light trigger, then utilize the light trigger. If an acoustic trigger would be more viable, then choose that. For the sake of simplicity, an acoustic trigger would help explain the mechanics of this protocol.

Initiation Phase: The initiation phase kicks off with all of the motes getting their distance values. All motes have their radio turned on waiting to receive a distance packet from the base station. The base station transmits the timer packet and the distance value is determined by the amount of time it takes the mote to receive the packet from the base station. Once that mote has received the distance packet, the mote stores the value and sends a reply packet back to the base station. The reply packet contains the Node ID and the distance value that was stored by the mote. The base station stores that information in a table. Once the node sends the reply packet then the nodes go into sensing phase. The node will go into the assimilation phase if no initiation packet is received before the specified timeout period in the initiation phase. Here it attempts to work itself back into the network seamlessly when that particular node’s power dissipates completely and power source is replaced.

Sensing Mode: Once assimilated or finished with the initiation phase, each node goes into sensing the environment. The radio is deactivated. Only the microphone and any other sensing module are activated during this phase. Once an environmental event happens, then the sensing node that captures the event will sound its buzzer. Once the buzzer sounds, the mote will then begin to transmit the data packet to the neighboring node. The data packet consists of the captured data, the distance value of the mote, and the original ID of the node capturing the data. Once the node transmits the data, it turns off the radio and goes back into sensing mode.

Data Reception Mode: When a sensing mote detects the buzzer trigger, it goes into Data Reception mode. In this mode the node switches on its transceiver and begins to receive any transmission. When the data packet is received, the mote compares the distance value from the packet and determines if the packet should be retained or dropped. If the distance value is greater than the mote's own distance value, then the packet is retained. Otherwise it will be discarded and the mote will go back into sensing mode. When a data packet is retained, the mote will begin the transmission process by sounding its buzzer to alert neighboring nodes to begin preparation for data packet reception. The mote with the data packet then begins to transmit the data packet. The distance value received from the original data packet is replaced with its own distance value. Afterwards, that mote shuts off the radio and goes back into sensing mode. This process continues until the base station receives the data.

Assimilation Phase: The assimilation phase takes place under two conditions. The first condition is when a node uses its entire power source, dies, and is subsequently furnished with a fresh power source. The second condition is when a new node is being introduced into the network. When a node is brought online it goes into the routine initiation phase, assuming the node is starting up with the whole network. The node will continue waiting for the initiation packet until the node reaches its timeout period. Once the node times out, it will move to the assimilation phase.

In the assimilation phase, the node assumes that it was once involved in the network. If this assumption is true, the base station will have a table of the distance values for that node in the network. The node can request that stored distance value. The node sends a Reboot RQST packet to the base station containing the Node ID. When the base station receives the RQST packet, the station finds the distance value based on the Node ID and sends the distance value back, using the Reboot ACK packet. Once the node receives the distance value then the node goes into the sensing phase. In the instance that the assumption isn't true and the node isn't included in the reference table, the base station will reply with a distance packet. The receiving node will then follow the same initiation packet steps and will reply with an ACK packet for the base station to receive and store in the reference table. The node will then go into the sensing phase.

IV. EXPERIMENT RESULTS

A. Experiment Materials

The mote used in this research is the MICA sensing platform. The MicaZ motes consist of an Atmel Atmega12LB microcontroller, a 51-pin expansion connection, and an IEEE 802.15.4 RF transceiver. The expansion interface board connects with the motes and contains the many different sensor packages. The current interface board being used is the MTS300 Muiltisensor Board. The sensor board contains a Light sensor, Sounder, Temperature sensor, and a microphone. The MIB510 Serial Interface Board is a board that provides the ability to download programs onto the MICA motes. It connects to the motes using the 51-pin connector and connects with the system containing the programmable software using a RS-232 Serial Port and is downloaded from the computer to each individual mote.

The MicaZ mote uses an Atmel Atmega 128L microcontroller and contains about 4kB of RAM and about 128kB of flash memory. The mote draws about 3.3 V to operate. Transmitting can draw about 17 mA and 8mA for an active controller state. Conversely, radio idling draws 20µA and 15µA is drawn when the controller in a sleep state. The transmission range can go up to about 100 m. The mote power source is 2 AA batteries.

The setup will consist of multiple nodes, and a base station. The nodes are randomly placed in the area of interest while the base station is at a predetermined location. The base station will consist of the MIB510 gateway board with a connected mote, and the data can be processed on the computer station. The nodes will just consist of the MicaZ motes, each with a sensor board for data gathering.

B. STER Limitations and Assumptions

There are some hardware limitations that prevent the potential utilization of this protocol. The hardware sensors on the motes are only able to capture intensity levels of the specific aspect of the environment. For example, the photo sensor is only able to record the intensity of the surrounding ambient light, not the wavelength of the light captured. This also goes for the accelerometer and microphone as well. The accelerometer only records whether or not the mote is mobile or not, but not necessarily what the relative location the mote is being moved to. The microphone only picks up the amplitude of the surrounding noise, and isn't able to make distinction with other qualities such as pitch, timbre, phase, etc.

The nodal hardware works well for applications when it comes to capturing environmental data. Unfortunately, it doesn't carry over as seamlessly for a communication triggering medium. Because of this, the method of communicating between nodes becomes very limited when using sensors as a triggering medium. The sensors equipped with the MicaZ capture temperature, light, speed, and sound data, and the hardware is only able to generate sound and light triggers. Each MicaZ node is also equipped with a MIDI buzzer on the sensorboard and has a pin connected to each LED that can be connected with a larger LED diode. So the two most feasible options using this hardware are using either a light triggering event or using an acoustic triggering event.

Choosing between the light and acoustic trigger implies consideration of and a trade-off between certain parameters. The first was the ease of node setup within the network. If light intensity is selected as the triggering event, the nodes would have to be setup such that the LED is bright enough and visible enough for the photoresistors to capture between each node. This would mean the placement of the nodes would have to be positioned such that the LEDs would have to be independently visible by each neighboring node. This is problematic because the LEDs would have to be visible, omnidirectional, and capable of generating light at a higher intensity level than that of ambient light in the surrounding environment.

Using an acoustic trigger instead seems to be a little bit simpler. Because sound is omnidirectional by nature, the placements of the nodes aren’t as constrained as the light trigger. The buzzer on the board prevents the need for adding any extra hardware. The spacing of the nodes should not exceed the range of the sound generated by the buzzer to the point that it won’t be drowned out by the ambient noise and not trigger the neighboring nodes. Because of the intangibles mentioned, it seemed that the acoustic trigger for this type of hardware would be a more feasible setup. Creating a communicating triggering medium, such as a dedicated IR transceiver or something similar may serve as a better medium with the software design.

A drawback to using this methodology is that using a sensor as a communication medium allocates that resource such that it will be hard or impossible to use that same sensor for data gathering. This is due to the fact that current sensor design does not provide a method of distinguishing between an environmental data capture event and a node triggering event. The general concept of the protocol is to make it flexible so that any sensor can potentially be the triggering sensor. If nodal hardware included a dedicated sensor or sensor port designed as a communication trigger between nodes then that will prevent the need for the tying up one of the environmental sensors. This would perhaps drive up cost of the node, but the tradeoff would be worthwhile given the reduced energy consumption of a signaling sensor like an IR transceiver when compared to the radio. The result would contribute to an overall increase in the lifetime of the network and a decrease in the number of man-hours needed for network maintenance.

C. Experiment Setup

The experiment setup tests the functionality of the protocol with respect to a sensor-triggering event and its interaction with neighboring nodes in the network. The setup consists of nodes, labeled Node 1 and Node 2. Both nodes are programmed with the node ID, with Node 1 having an ID of 1 and Node 2 having an ID of 2. The experiment is setup in a controlled environment with no ambient interference and with no physical objects that dampen or block sound waves. The experiment uses the same setup discussed in section 4.1. The nodes are sensing for both a nodal triggering event and an environmental triggering event. The environmental trigger occurs when the light sensor detects a low level of light in the observed space. The nodal triggering event is an acoustic event generated by the nodes. When a positive environmental event occurs the node will generate a nodal event using its buzzer and transmits its Node ID. The receiving node will compare the Node ID with its own id and if the value is higher the node will generate its buzzer and transmit. If the value is lower the yellow LED turns on to dictate that the transmission was received but was not retransmitted. So in other words, if Node 2 creates the nodal triggering event then Node 1 should detect that, compare values, generate the nodal trigger and retransmit. This should not be the case if Node 1 captures the environmental trigger event. Instead, after Node 1 generates the nodal trigger and transmits, Node 2 should flip on the yellow LED to dictate that the value is not a valid Node ID value.

The experiments are setup with 3 different scenarios. The first scenario is to determine the maximum distance neighboring nodes can be apart and function properly. The method to scenario one is to move one sensor back further while the other is stationary and create a triggering event. The threshold distance is the furthest distance apart where the nodes are able to trigger each other. Once this is determined, the second and third scenario will be setup.

Scenario 2 and 3 introduce ambient acoustic interference into the experiment. The source of the interference will be placed in the location of the receiving node. Scenario number 2 introduces ambient noise that is a constant frequency. The objective of scenario 2 is to observe how loud the ambient noise must be before the nodes become ineffective in communication. The communication between nodes is considered ineffective when a receiving node registers a false positive or when the neighboring node isn’t able to capture the nodal trigger event due to the intensity of the ambient noise. The frequencies used are 440 Hz and 2.93 kHz for each trial and the intensity is measured in decibels. Scenario 3 observes the same functionality. The difference between scenario 3 and 2 is that scenario 3 introduces varying frequencies that are more natural to the real world by playing music in the background.

When completing scenario 1, the nodes were able to trigger each other while being about 19 feet apart. Anything past that and the response between the nodes when triggering starts to become inconsistent. Scenario 2 has two separate trials. The first trial is with the ambient noise generating at 440 Hz. During this trial, results depict that the receiving node can register the nodal triggering event all the way up to the maximum volume output of the background source. The decibels generated for trial 1 averaged around 79 dB. The second trial generates noise at 2.93 kHz. The results show that the nodes become ineffective at about 80 dB at that frequency.

Scenario 3 replaces the static ambient noise to a more realistic ambient noise with varying frequencies. Four different trials were run using different genres of music at maximum volume of the background source. Trial 1 was run using a Metallica's "Fixxxer" with the decibels ranging from 67-74 dB. Trial 2 was run with Dirty Money's "Hello Good Morning", and the decibels ranged from 70-78 dB. Trial 3's background music was "Monster" by Kanye West. Trial 3's decibel ranged from 75-84 dB. The final trial was Beethoven's "Moonlight Sonata" and the decibel ranged from 86-90 dB. In all trials, the nodes were able to consistently trigger each other. There were no false positives in any of the setups.

D. *Energy Consumption Calculations*

Energy consumption calculations are based on (12) and (13) to determine the amount of energy consumed at a node level and network level as well. The LEACH, STER, REDRP, RPTAW, and SPEED protocols were observed and compared against each other to provide some insight into their respective energy consumption performances. In order to create the same conditions for each protocol, the constants needed for each protocol simulation will be as close to consistent as possible. Each network will consist of 8 nodes. Based on Heinzelman's experiments, P_{R0}, P_{T0}, ε, η, and α will be the same constants [15]. The transmission range will generally be set at 10 meters unless stated otherwise. At that range, the transmission has to be -15 dB. Also, to remove the bias of data packet sizing of different protocols, the experiment assumes that only1 bit is needed to send or receive the necessary data at the rate of 1 bit/ µsec. Making this assumption isolates the overhead of each protocol with respect to data transmission. Table 6 shows the constants that Heinzelman used.

Table 6
Power Consumption Model variables

variables	Values
P_{R0}	59.1 mW
P_{T0}	26.5 mW
H	35%
A	2
E	0.0005
d_h	10m
d_s	20m

For cluster-based topologies, it would make sense that the distance from cluster member to cluster head is shorter than the distance from cluster member to the base station. If each cluster member is equidistant from the cluster head, then the cluster head has to be at least twice as far from the base station as the closest cluster member. Just for the sake of simplicity, if each cluster member is 10 meters away, then the distance from the cluster head to the base station is at least 20 meters away. The variable d_h denotes the distance that lies between the member cluster and the cluster head. The variable d_s denotes the distance from the cluster head and the base station. These distance values are the average distances that the cluster heads will be from the cluster members and the base station respectively. Note that in the real world the distances will fluctuate due to the fact that there will be rounds where a cluster head is spatially biased. Some nodes will need more power to transmit packets to the cluster heads and closer members will need less. This would also apply for cluster heads when they would need to transmit across the cluster or to the base station.

LEACH Energy Consumption: LEACH is a cluster-based protocol and has to go through two phases to transmit data from the source to sink. The first phase is the setup phase. During this phase, the cluster head has to setup the TDMA schedule for the cluster. All nodes in the cluster are active. The cluster head transmits the schedule to the cluster members and each cluster member receives the schedule for 1 μsec. When in the data transfer mode, each member is on only when it needs to transmit its data. It then goes to sleep until the next round. The cluster head, however, has to be on for the whole duration. Thus, 7 μsec is needed for reception at 10 meter, and 1 μsec is required for each cluster member to transmit data to the cluster head. The cluster head also transmits to the base station across 20 meters for 1 μsec. Table 7 shows the power consumed during the setup phase for the cluster head and member while table 8 shows the amount of energy used during the data transfer phase.

Table 7
Local Power and Energy consumption for LEACH nodes: Setup Phase

Node Role	Power Used(mW)	Energy Used (μJ/bit)
Member	85.74285714	0.086171429
Head	85.74285714	0.113242857

Table 8
Local Power and Energy consumption for LEACH nodes: Data Transfer Phase

Node Role	Power Used(mW)	Energy Used (μJ/bit)
Member	26.64285714	0.026642857
Head	86.17142857	0.440771429

At a local level the cluster head takes the brunt of the load using this methodology. Moving forward with modeling STER using the power consumption model, the overall performance based on (10) and (11) can be compared for LEACH and STER respectively.

STER Consumption Model: The proposed protocol design, STER, is a reactive protocol that takes a slightly different approach from LEACH. The amount of energy dissipated hopes to be smaller than that of any other protocol. When applying the power consumption model to STER, the parameters are similar to LEACH, with a couple of exceptions. The first exception is that the nodes will all be equidistant from each other. The second exception is that all the nodes will be receiving and transmitting 1 bit at 1 µsec. The power is determined using the variables in Table 6. (10) gives the energy output with the transmission distance of 10 m. The energy is determined by taking results from (10) and using those results as inputs into the corresponding energy equation with t = 1 µsec. Table 9 shows the amount of power and energy consumed respectively.

Table 9
Local Power and Energy consumption for STER nodes

Node Role	Power Used(mW)	Energy Used (µJ/bit)
Member	85.74285714	0.086171429

The energy consumption at a local level slightly outperforms LEACH on a local level against both the cluster member and cluster head. Although at a local level STER looks more energy efficient, observing the overall network consumption would help give better insight in regards to how well the protocol performs.

REDRP Consumption Model: REDRP is the reactive protocol that most closely relates to STER. REDRP uses a dynamic routing system that allows on-the-fly route discovery and route changes in case the discovered route becomes broken. The initiation phase consists of all nodes being in sensing mode and listening for radio packets as well. During the route discovery phase, the discovery node transmits discovery packets to the neighboring nodes. Upon reception, the receiving nodes will reply back to the discovery node. Based on the feedback from the receiving nodes, the discovery node will choose the best node to select for data propagation. The node chosen by the discovery node to be the next hop will receive another packet indicating its selection as the next data destination point. The receiving node repeats the process as it becomes the node in discovery mode. This process continues until the sink is reached. The sink then sends a confirmation packet. This packet propagates down the path in reverse order until the source is reached. Lastly, the data is transmitted from the source to the sink via the confirmed route channel. The route rediscovery step occurs when one of the nodes that have been confirmed as part of a route path is triggered by a separate interrupt due to a sensing event or route discovery packet from a separate node. The rediscovery phase will be ignored to reduce complexity of the model.

To keep the model simple and consistent with the other protocols, the variables in Table 6 determine the energy output at a distance of 10 m. Also, no conflicting route discovery jobs from a separate route discovery nodes will be considered. The following table displays the local level energy consumption of the nodes.

Table 10
Local Power and Energy consumption for REDRP nodes

Node Role	Power Used(mW)	Energy Used (µJ/bit)
Member	85.74285714	2.203857143
Source	85.74285714	2.149714286

The nodes consume more energy than the previous protocols due to the fact that the nodes are always in receive mode. However, if the node receives data over the air from a neighboring node or it captures significant data that needs to be forwarded, the receive mode is exited. In those cases the node will then transmit the data to the neighboring nodes to setup a route for data forwarding. Also, because it takes so long to build the route through requests, acknowledgements, and confirmations between nodes, the nodes spend a significant amount of time idling before sending the actual data.

SPEED Consumption Model: SPEED is another reactive protocol that is designed to provide the fastest and shortest route to the sink for both stationary and mobile nodes. SPEED uses an assortment of different systems to accomplish its goal. Some systems include a feedback loop system, a beacon exchange system and a SNGF system. The beacon exchange system requires a node to provide time-based transmission bursts to update neighboring nodes of its location using beacon packets. This requires nodes to listen constantly to maintain current node positions of so that each node can dynamically learn which route is the best at that time. During transmission of data, the nodes use the latest information about the network from the beacon packets to provide the best route.

Staying with the trend, Table 6 variables are plugged into (10) with the distance of 10 m to model this protocol. The energy of a node is reflected in the following Table 11.

Table 11
Local Power and Energy consumption for SPEED nodes

Node Role	Power Used(mW)	Energy Used (µJ/bit)
Member	26.64285714	27.48512857

The SPEED algorithm uses quite a bit of energy per node. This is due to the radio constantly sniffing for data packets, even after it receives and forwards data. SPEED is not particularly trying to be as energy efficient as possible, but is instead attempting to have the highest QoS possible for a large network mobile network.

RPTAW Consumption Model: RPTAW is a reactive protocol that uses clustering to control the flow of data from the source to sink of a network. The idea is similar to LEACH. Therefore, each set of nodes defines a cluster head and a set of members in a cluster. The difference between LEACH and RPTAW is that the task of the cluster head in LEACH is split up between the head and what is called a Relay Node. The relay node is responsible for sending data packets to the sink via a direct path or via other relay nodes from neighboring clusters. The cluster head is only left with aggregating the data from the member nodes, and transmitting that to the relay node for data forwarding to the sink. The member nodes only have to worry about transmitting data when it is time for them to send their respective data. The rest of the time the node can sleep until the round expires. However, the node has to wake up in time to receive the new schedule from the new cluster head. The cluster head and the relay node generally have to always be in the receive mode until they need to send data. In the election phase, the cluster head only has to transmit to the member nodes the TDMA schedule and to the node that it assigns to be the relay node. The other nodes just have to receive the data from the cluster head. Table 12 shows the energy breakdown for a node in the election phase, and Table 13 shows the energy consumption for the data transfer phase.

Table 12
Local Power and Energy consumption for RPTAW nodes: Election Phase

Node Role	Power Used(mW)	Energy Used (μJ/bit)
Member	59.1	0.0591
Head	26.64285714	0.026642857
relay Node	59.1	0.0591

Table 13
Local Power and Energy consumption for RPTAW nodes: Data Transfer Phase

Node Role	Power Used(mW)	Energy Used (μJ/bit)
Member	26.64285714	0.026642857
Cluster Head	85.74285714	0.381242857
Relay Node	86.17142857	26.99745714

This protocol has the highest energy consumption disparity between nodes. While the cluster head only has to aggregate the data, the relay node has to receive and transmit on demand across a longer range. This design reflects the biggest drawback in cluster-based networking systems, overloading a single node in exchange for lessening the load of the rest of the cluster.

Total Network Consumption: With the scope localized to just a single node, it would seem that some protocols are less efficient than others. Understanding the basic principle of wireless network design, nodes will eventually consume all of its power and will no longer be able to perform within the network. The different strategies of different protocols attempt to prolong the inevitable and in essence maintain the utility of the network, even as nodes start to become ineffective. Because of this and to get a better idea of how a protocol affects the network, the scope must be broadened to observe all nodes within the network.

Table 14 displays the amount of energy dissipated below by the each network as a whole. Each protocol dissipation value per round is determined by summing up all of the nodes for a particular phase. Each protocol has various tasks for nodes and various amounts of nodes assigned to that particular task, so the calculations take into consideration the multiple complexities for each system. Table 14 results are as follows:

Table 14
Network Energy consumption per round

Protocol	Network Energy per round (μJ/bit)
LEACH	1.343714286
STER	0.689371429
SPEED	219.8810286
RPTAW	27.84568571
REDRP	17.57671429

STER outperforms the other protocols, followed by LEACH, RPTAW, and finally SPEED. Constant sniffing for packets seems to the major contributor to the amount of energy that is consumed. SPEED has more nodes tasked with sensing because of each node needing to be location aware as nodes can potentially change position. In contrast, STER and LEACH use no node or only one node tasked with having its radio in use for an extended period of time. The results reflect a direct correlation between radio use and energy consumption.

STER multi-hop versus single-hop topology: STER's design follows suit with most other protocols in data propagation by using the multi-hop method to pass data from source to sink. Previous calculations illustrate the amount of energy expended using the multi-hop method. The energy dissipated for the network can be compared to the amount of energy consumed using the single-hop method. With the single-hop method, source nodes will directly transmit the data to the sink. The following table reflects the amount of energy that is consumed for each if each node used the single-hop method to propagate data. The data correlates to how much the network expends if a node has to forward data. Since the nodes involved is between the source and sink. Since the sink has unlimited power, the amount of power directly correlates to the power consumed by the source. The same setup parameters will be used to calculate the amount of energy. All 8 nodes will be equidistant at 10 meters apart. For example, node 8 will have to transmit collected data 80 meters to the sink, while node 2 will only have to transmit 20 meters.

Table 15
STER Energy Consumption using Single-Hop method from Source to Sink

STER Single-hop Energy			
Node	Distance	Power Used(mW)	Energy Used (µJ/bit)
1	10 m	26.64285714	0.026642857
2	20 m	27.07142857	0.027071429
3	30 m	27.78571429	0.027785714
4	40 m	28.78571429	0.028785714
5	50 m	30.07142857	0.030071429
6	60 m	31.64285714	0.031642857
7	70 m	33.5	0.0335
8	80 m	35.64285714	0.035642857

Results show that there is less energy expended using the single-hop method when compared to using multi-hop method for STER. Even the node that is furthest from the base station uses less energy than the original multi-hop design. There are some systemic biases however. The first bias is that the further a node is away the base station the more energy is consumed to transmit data when compared to the node that is closer. Using the single-hop method doesn’t provide the same amount of balance on the network as the multi-hop method. The other bias is that the further a node is the more transmission power is needed. More transmission power raises the probability of interference within the network. Because of these biases it would be more efficient for the network to transmit at lower transmission range.

E. Discussion

As the results reflect, reducing energy consumption can be achieved by reducing the amount of radio use. The radio is the biggest culprit for drawing the most current according to multiple hardware data sheets. It would make sense that the longer the radio remains in the active state, the faster the node dies. This is due to an increase in the network's energy depletion rate. Network topologies do have their pros and cons. Directed diffusion offers more independent responsibility for each node to forwarding data. This method generally results in the load being distributed uniformly amongst the network. The drawback for directed diffusion is that even though the design of directed is to spread out the load evenly throughout the network, there is a systemic bias with nodes. The nodes closer to the sink node would be utilized more often than the other nodes further away from the sink. Clustering resolves the issue that directed diffusion has by having dedicated nodes aggregate data and pass it on to get processed. The drawback for clustering, however, is that generally a node is typically tasked with a bigger load that may include transmission times to be larger than other nodes. While the five protocols observed shows that more energy is consumed on average by diffused topologies, if the design works to minimize radio usage, the energy can rival the cluster type protocols.

The results show a good indication of how a protocol should perform based on their behavior. Results also show the variance in which proactive and reactive protocols would perform in comparison to each other. Unfortunately, these performance numbers only represent ideal situations and therefore they are not indicative of the situation that is most common. Additionally, there are other factors affecting the performance of the network. Factors include environmental elements like physical obstruction, ambient noise and light, dropped or corrupted packets, latency, etc. Because of real-world elements, performance may never live up to what the performance numbers.

Even though some protocols perform better than others, one protocol may be the better option than others. This would depend on the application. Some applications may require WSNs to provide higher temporal resolution, while other applications may want a system that provides real-time data. Even though on average reactive protocols are more energy efficient than proactive protocols, proactive or hybrid protocols may be the way to go if the application requires higher temporal resolution. However, if there is only a need to get specific data passed through, reactive protocols would be more fitting. If there was a need to monitor the amount of force generated in an earthquake, SPEED would be a better choice than the other 4 four protocols observed just because of its ability to perform even while nodes are mobile. STER wouldn't be as effective as SPEED in the earthquake scenario because STER isn't equipped to handle nodal mobility, whereas SPEED has that capability. However, STER would definitely be a better choice over SPEED if there was a need to monitor a room for significant temperature changes because it can capture the data while being more energy efficient.

A big plus for majority of protocols is that they aren't dependent on a specific hardware model and can be implemented on any node designed to work as a wireless sensor network. STER is no different. With that said, it is almost necessary to have some hardware platform to work with. So while some hardware is required, none of the protocols observed are constrained to any hardware device. As long as the hardware supports interfaces to an onboard radio and a set of nodes, it should be able to support most protocols. With STER relying more on sensors than other protocols, it would have to require at least two separate sensors, as one would be exclusive to communication triggering. Also, STER would require the ability to generate the same kind of energy that one of the sensors detects. A selling point for STER is its leniency in the sensor selection process. Therefore, it is not dependent on a particular sensor. It allows the triggering sensor for communicating between nodes to be catered towards what it best for the application. As long as there is a way to create a trigger that is detectable by neighboring nodes, the protocol can be tweaked to potentially have any kind of trigger. Some triggering options can be acoustic, light, or any other kind of sensing, as long as there is a way for the node to create and sense the same kind of energy.

The downside for STER is that the current sensor hardware set has been designed to sense the environment not as a communication tool. This simple element limits the potential for what STER can do because it utilizes a component that isn't optimized for communication. All the sensors only measure intensity, and no other dimension, so it would be difficult for the sensor to detect different pitches of sound, or wavelengths of light. Due to this limitation it's hard to differentiate between an environmental trigger and node trigger. This results in the improbability of a sensor to functioning as both a data gathering device and a node triggering device. Because of this conflict, a sensor must be exclusively dedicated to creating that communication trigger. If there was a sensorboard design that included a dedicated communication triggering medium or sensor designs capable of detecting and reproducing more than one-dimensional measurements, then STER and similar design protocols using sensor triggering as a form of initiating communication would be more viable.

Because this is the first idea of alternate WSN communication, there is a lot of upside for this kind of method. The limitations with the hardware are apparent. However, with more research stimulating the basic concept of using sensors as communication medium, WSNs can potentially vastly improve energy dissipation of the system. Challenges such as restriction to line of sight configurations, ambient noise, and general interference would arise, so some kind of filtering would be necessary to prevent false positives from occurring. More dynamic sensors could potentially help to differentiate the false positive anomaly. Another possibility for reducing false positive triggering is to include a sound generating device with the ability to generate different frequencies. Currently the buzzer is only capable of emitting a 4 kHz tone and the microphone is only able to detect that one frequency. If custom frequency generation is coupled with the microphone’s ability to set and capture specific frequencies, false triggering events due to ambient noise in the environment can be significantly reduced. Unfortunately, this may result in higher monetary cost per node due to the integration of more sophisticated sensors on the board. All in all, sensor triggering seems to be a promising new approach to minimizing energy for WSN systems.

V. CONCLUSION

If one was asked if they were aware of wireless sensor networks and their existence, most would probably be unaware of the technology’s existence. Contrary to this presumption, WSNs have already been subtly assimilated into many aspects of our industrial lives. Implementations range from environmental and medical industries, to military and security applications. Even though the technology is already being taken advantage of, it is still relatively new and still shows rooms for improvements. There are many points of concern for WSNs, including QoS. The more exclusive and relevant issue of WSNs is their network sustainability. Network sustainability subsequently impacts the QoS negatively because once key nodes in the network deplete their energy sources the network can become less effective. Due to the impact that network sustainability has on the performance of the network, a significant number of protocols are designed with energy optimization in mind over other network issues.

Based on the power consumption model, results show that using STER uses the least amount of energy per round when compared to other protocols of different variations and techniques. This shows that at the very least, there is quantitative validation for deeper investigation into the viability of sensor-based communication between nodes versus using radio communication. This concept has the potential to significantly impact the way WSNs are designed and result in considerable reductions in WSN energy consumption. There are challenges that arise with the concept behind STER. However, with simple hardware design modifications and sensor choice flexibility, STER is a frontrunner of a concept that has tremendous upside.

REFERENCES

[1] He, T., Stankovic, J. A., Lu, C., and Abdelzaher, T., *SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks*.2003.

[2] Toscano, O. M. and Bello, L. L., *An Energy-efficient Real-Time Communication Framework for Wireless Sensor Networks*.2001.

[3] Wang, Y., Lin, Y., Fu, P., and Tsai, C., *REDRP: Reactive Energy Decisive Routing Protocol for Wireless Sensor Networks*. UIC 2006, LNCS 4159, pp. 527 – 535 2006.

[4] Yassein, M. B., Al-zou'bi, A., Khamayseh, Y., and Mardini, W., *Improvement on LEACH Protocol of Wireless Sensor Network (VLEACH)* International Journal of Digital Content Technology and its Applications 3(2), 2009.

[5] Chee, Y., Rabaey, J.M. and Niknejad, A.M., *A class A/B low power amplifier for wireless sensor network*", ISCAS 4, pp. 409-412 2004

[6] Mohajerzadeh, A. H., Yaghmaee, M. H., *An Efficient Energy Aware Routing Protocol for Real Time Traffics in Wireless Sensor Networks*. Ultra Modern Telecommunications – ICUMT, pp.1-9 2009

[7] “MICAZ Wireless Measurement System” Crossbow http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf

[8] Wang, Q., Hempstead, M., and Yang, W., *A Realistic Power Consumption Model for Wireless Sensor Network Devices*, IEEE 1(1), pp: 286 – 295 2006.

[9] “CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver”, Chipcon 2004.

[10] Kramer, M., *Energy Measurements for MicaZ Node*, University of Kaiserslautern, Germany 2006.

[11] Akyildiz, I.F., Su W., Sankarasubramaniam Y., Cayirci, E., *Wireless Sensor Networks: A Survey*. Computer Networks 38, pp. 393-422 2002.

[12] Ye, F., Zhong, G., Lu, S., *PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks*. IEEE International Conference on Network Protocols, 2002.

[13] Liang, W., Yu, H, Zeng, P.,Che, C., *BESM: A Balancing Energy-aware Sensor Management Protocol for Wireless Sensor Network*. International Journal of Information Technology 12(4), 2006.

[14] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H., *Energy-efficient communication protocol for wireless micro sensor network*”, IEEE Proceedings of 33rd Annual Hawaii International Conference on System Sciences 2, 2000

[15] Mohajerzadeh, A.H. and Yaghmaee, M.H.,“*An efficient energy aware routing protocol for real time traffic in wireless sensor networks*, ICUMT, pp. 1-92009

[16] Lindsey, S. and Raghavendra, C.S., *PEGASIS: Power-efficient gathering in sensor information systems*, IEEE Aerospace Conference Prcoeedings 3, pp. 3-1125 -3-1130, 2002

[17] Ye, W. and Heidemann, J., *Ultra-Low duty cycle MAC with scheduled channel polling*, SenSys - Conference On Embedded Networked Sensor Systems, pp. 321-334, 2006

[18] Mica Sensor Board Review: http:// webs.cs.berkeley.edu/weekly/review

[19] Akyildiz, I. and Vuran, M.C. 2010. Wireless Sensor Networks. Wiley.