SELF-SELECTING RELIABLE PATH ROUTING FOR ALL ENVIRONMENTS USING SENSE WITH VISUALIZATION

By

Thomas Adam Babbitt

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved:

Boleslaw K. Szymanski, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

February 2009
(For Graduation May 2009)

CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to thank my wife Bridget and children Grace, Margaret, Eleanor, and Thomas for their strength and encouragement throughout my many long days at RPI. I would also like to thank Professor Boleslaw Szymanski whose expert advice and constant support was instrumental to the success of this thesis and numerous other projects and papers on which we collaborated. Thanks are also due to Chris Morrell; I could not have asked for a better research partner. Finally, I am very appreciative to Joel Branch, Kamil Wasilewski, Sahin Cem Geyik, and Wang Zijian for their assistance and contributions to this research.

ABSTRACT

Routing protocols for Wireless Sensor Networks(WSN)face three major performance challenges. The first one is an efficient use of bandwidth that minimizes the transfer delay of packets between nodes to ensure the shortest end-to-end delay for packet transmission from source to destination. The second challenge is the ability to maintain data flow around permanent and transient node or link failures ensuring the maximum delivery rate of packets from source to destination. The final challenge is to efficiently use energy while maximizing delivery rate and minimizing end-to-end delay.

Protocols that establish a permanent route between source and destination, such as Advanced On Demand Vector Routing (AODV), send packets from node to node quickly, but suffer from costly route recalculation in the event of any node or link failures. Protocols that select the next hop at each node on the traversed path, such as GRAdient Broadcast (GRAB), Self Selective Routing (SSR), and Self Healing Routing (SHR), suffer from a delay required to make such selection. This led to Self Selecting Reliable Path Routing (SRP), which attempts to take advantage of both by creating a reliable path.

Even with the use of a reliable path the way in which a protocol repairs routes determines the number of packets lost by each failure and ultimately affects the energy used for communication. This thesis presents a novel family of wireless sensor routing protocols, the Self-Selecting Reliable Path Routing Protocol Family (SSRPF), that address all three of the afore-mentioned challenges. In addition to collaborative work on the SRP protocols, which make up two-thirds of the SSRPF, the specific contributions of the author of this thesis were modifications of the route repair procedure of the protocol and investigation of the impact that the choice of route repair has on the overall performance. These improvements are the basis of the third protocol in the SSRPF, Reliable Path Self-Selecting Protocol (RPSP).

1. Introduction

Wireless sensor networks consist of a large number of nodes each with a radio transmitter for wireless communication, a receiver for sensing and receiving transmissions and a CPU for processing applications and protocols. Many wireless sensor networks consist of unattended battery-powered nodes. These autonomous networks must be fault-tolerant and energy-efficient in all aspects of their operation. These properties are critical for routing, since multi-hop communication is fault-prone as well as energy-intensive. Commonly observed in such networks are faulty (or, potentially subverted) nodes and transient and asymmetric links caused by wildly oscillating packet reception quality. Faulty nodes and transient links cause severe packet loss and spontaneous network topology changes [1, 2]. Radio operation is typically the most costly function in wireless sensor nodes, as evidenced by a study in [3] and typical hardware specifications given in [4, 5].

The traditional approach to multi-hop routing uses a routing table that indicates the neighbor where a packet is to be forwarded to reach a destination; some prominent examples include AODV [6] and Directed Diffusion [7]. This fundamental approach emulating traditional wired network protocols requires nodes to constantly maintain an updated routing table that includes individual neighbor's states (e.g., active or sleeping). In typical wireless sensor networks operating conditions, this approach requires significant overhead to maintain a usable routing table, especially if fault-tolerance is to be supported. Hence, providing efficient routing protocols that naturally accommodate and perform well in fault-prone conditions is still an open and formidable challenge.

Different applications and nonstandard hardware of WSNs result in the diverse network environments in which they operate. Generally the exact location of a node is not planned and they are scattered throughout their operating environment. This often leads to either entire networks or portions within a network having extremely high or very sparse node density. Hence, WSN routing protocols must maintain performance in networks that have both a dense and sparse dispersion of nodes.

The terrain and harshness of the climate in which a WSN is employed, determine how likely nodes will either fail completely or will experience intermittent node and link failures. If the location is remote or behind enemy lines, the ability for those nodes to be quickly replaced or repaired might be significantly limited. Since WSNs can be employed in all operating environments, a routing protocol must perform well regardless if there is a high rate of permanent failures or a high rate of transient node or link failures, or both. The applications purpose and its ability to recover from lost or duplicate data packets determine how essential the data delivery rate is. Three major challenges need to be addressed while designing WSN protocols able to perform in all operating environments.

The first challenge is to efficiently use bandwidth to minimize the end-to-end delay in packet transmission. Traditional wired approaches such as AODV [6] and Directed Diffusion [7] do a good job of quickly forwarding packets especially when the network has a low rate of node or link failures; however, when this is not the case,then either packet losses uncontrollably increase or a costly repair routine is frequently evoked. The second challenge is to maintain a high delivery ratio even in the face of node or link transient or permanent failures. Protocols that determine the next forwarder at each hop work well even with high rates of node and link failures because they are memory-less. Some examples of protocols that fall into this category are SSR [8, 9, 10], SHR [11, 12], GRAd [13], and GRAB [14]. The final challenge is to both efficiently use the bandwidth and maintain dataflow while minimizing energy use. Since radio operations are the most energy consuming operation performed by a node. The number of nodes in sleep mode and the number of broadcasts necessary to either forward packets or maintain route information determine jointly the energy efficiency of the protocol.

This thesis presents a novel family of wireless sensor routing protocols, the Self-Selecting Reliable Path Routing Protocol Family (SSRPF) which was inspired by the family of Self Selective Routing(SSR) protocols [11] and address all three challenges listed above. There are three protocols in the family. All three were extended from the SSR protocol. The first is Self Selective Reliable Path Protocol (SRPv1) [15] which finds a reliable path by cutting the back off delay of a winning node, ensuring its future selection, thereby expediting transmission of packets from source to destination. The second is Self-Selecting Reliable Path Protocol (SRPv2) [16] which, compared to SRPv1, modifies the route repair routine by not changing the hop count at the node level. The final protocol is the Reliable Path Self-Selecting Protocol (RPSP) which modifies the route repair routine to eliminate the lost packets that occur in the repair routine for SRP and is the major contribution of this thesis.

This thesis discusses a novel route repair routine used by RPSP. This route repair routine both avoids losing packets, as was the case in SRP, and reduces the number of transmissions needed to repair a route. This repair routine both decreased the end-to-end delay and the amount of packets broadcast which reduces the amount of energy used.

There exists numerous other protocols that, like RPSP, route attempting to avoid creating a routing table and let receiving nodes contend for forwarding packets. However, many require geographical location information, which RPSP does not. Three such protocols, GRAd [13], GRAB [14], and BLR [17] do not have a route repair routine. GRAB uses a more aggressive fault-tolerance technique allowing multiple paths to a destination. RPSP relies strictly on its prioritized transmission back-off delay technique to support (limited) fault-tolerance. Other protocols, such as, GeRaF [18], IGF [19], PSGR [20] and SIF [21] use eligibility regions for packet forwarding requiring detailed knowledge of geographical placement of currently active nodes. This creates the same issue as a routing table which is difficult to obtain and maintain in wireless sensor networks.

All simulations for all protocols contained in this thesis were conducted using SENSE. SENSE is an easy to use extremely robust simulation tool; however, there is not a visualization tool provided with it. A simulation tool intended for use with ns2 [22], was discovered. This tool, entitled iNSpect [23], was written by a group of researchers at Colorado School of Mines. This tool in conjunction with SENSE provides the ability to easily create a playback of a simulation in SENSE. This animation written about in [24] was used to ensure that the route repair routine in RPSP worked properly.

The remainder of this thesis is organized as follows. Chapter 2 describes SENSE and gives some backgound into why it is the correct choice for use in WSN simulations. Chapter 3 is the historical work on the Self Selecting Routing (SSR) Protocols. Chapter 4 presents the collaborative work between Chris Morrell and the author mainly the first two members of the SSRPF family of protocols SRPv1 and SRPv2 [15, 16, 25]. Chapter 5 presents the primary contribution to this research topic in the form of the newest SSRPF protocol, Reliable Path Self-Selecting Protocol (RPSP). In addition, chapter 5 presents comparisons between SRPv1, SRPv2, AODV, and RPSP and the use of the simulation tool to ensure that the route repair routine works properly. Finally, Chapter 6 presents conclusions and possible future work that could stem from this research.

2. SENSE

SENSE is designed to be an easy to use, efficient and powerful sensor network simulator written in C++ that was originally presented in [26]. Three critical factors were taken into account when building SENSE. They were extensibility, reusability, and scalability. The enabling force behind the fully extensibility network simulation architecture is the progress made on component-based simulation. SENSE introduced a component-port model that frees simulation models from interdependency usually found in an object-oriented architecture, and then uses a simulation component classification that naturally solves the problem of handling simulated time.

The component-port model, built on top of COST [27] and CompC++ [28], makes simulation models extensible: a new component can replace an old one if they have compatible interfaces, and inheritance is not required. The simulation component classification makes SENSE extensible allowing advanced users the freedom to develop new variations of SENSE that meet their needs by modifying low level components such as layers of the protocol stack, mobility, and power management.

The removal of interdependency between models also promotes reusability. A component developed for one simulation can be used in another if it satisfies the latter's requirements on the interface and semantics. There is another level of reusability made possible by the extensive use of C++ templates: a component is usually declared as a template class so that it can handle different types of data.

There are two types of ports in SENSE. The first are inports that are functional in nature and implement a certain function. The second are outports that are abstractions of a function pointer. They are the definitions of functionality for others. The ports connect components. In the protocol stack, where each layer is a component, there would be a inport from the layer below and an outport to the layer above. The ports are used to transfer data and management information between components.

Unlike many parallel network simulators, especially SSFNet and Glomosim, parallelization is provided as an option to the users of SENSE. This reflects the belief that completely automated parallelization of sequential discrete event models, however tempting it may seem, is impossible, just as automated parallelization of sequential programs. Even if it is possible, it is doomed to be inefficient. Therefore, parallelization models require extra effort than sequential models, but a good portion of users are not interested in parallel simulation at all. In SENSE, a parallel simulation engine can only execute components of compatible components. If a user is content with the default sequential simulation engine, then every component in the model repository can be reused.

SENSE provides the ability to easily expand to support newly developed protocols and was written to include support for many popular protocols. The protocols included range all the way from the physical layer to the application layer, and include IEEE 802.11, AODV, several radio models, several power management models, and others. SENSE was created as a simulation tool for wireless sensor network in response to the network simulator ns2 [22].

Without many add-on packages, ns2, originally written as a wired network simulator, is not well suited to simulate wireless sensor networks. Since its first publication and use in simulating SSR, SENSE has gained a following in many places throughout the world. Conducting a Google search on "simulating results of WSNs," reports more than 36 papers. For example, in [29] the authors simulate C^2E^2S (Cluster and Chain based Energy Delay Efficient Routing Scheme) for wireless sensor networks. In [30], the authors used SENSE to simulate Coordinate-based Data Dissemination protocol (CODE) and Sink Cluster-based Data Dissemination protocol (SIDE). In addition to these, SENSE was used as the simulator by [31] and [32].

3. Self Selective Routing Protocols

Because wireless networks use broadcast communication, there is a fundamental difference between it and wired networks which use point-to-point communication methods. Wireless networks are also limited to half-duplex because a single receiver can only broadcast or receive on a single frequency. Wireless networks can take advantage of the broadcast communication through self-selection [33] to determine a node possessing a desired property enabling it to better succeed at forwarding a message. By employing a prioritized transmission back-off delay, each node can compete to determine which one has the best chance to forward a message. Upon self-selection, that node becomes the next link in the transmission of data. The ability of a node to self-select in order to forward a packet is the basis of all protocols in the SSRPF family.

The SSRPF contains a number of protocols with different end-to-end delay, end-to-end throughput and energy efficiency levels. Each of these protocols take advantage of self-selection by employing a prioritized transmission back-off delay. This chapter introduces the original research into self-selective routing that is written about in [16]. It discusses the Self Selective Routing (SSR) and Self-Healing Routing (SHR) protocols. Subsequent chapters will discuss the Self-Selective Reliable Routing Protocol (SRP) and Reliable Path Self-Selecting Protocol (RPSP).

3.1 Self Selective Routing (SSR)

There are two stages in SSR. The first is a discovery stage and the second is the transmission stage. In all protocols of the SSR family, each node knows its distance, in terms of the number of hops, from a destination node. This distance is established via an initial route request and route reply stage. In this thesis, we assume static (non-mobile) nodes, so hop distances of a node to any other node can only be changed by node or link failures. For the packet forwarding process, instead of only one designated neighbor receiving the packet sent by the sender, all of its neighbors receive it. The neighbor nodes then use the self-selection algorithm to decide autonomously which node will forward the packet. This self-selection algorithm uses a prioritized transmission back-off delay scheme. In this scheme, after a node receives a packet, it sets a timer for a random delay based on its distance, in terms of hops, from the destination. The transmission back-off delay for SSR is specifically determined by the following equation:

$$d_{back-off} = \begin{cases} \lambda \cdot ((h - h_{expected} + 1) \cdot U(0,1) & if h > h_{expected} \\ \frac{\lambda}{h_{expected} - h + 1} \cdot U(0,1) & if h \leq h_{expected} \end{cases} \tag{3.1}$$

h is the node's hop distance from the destination, $h_{expected}$ is the sender's hop distance minus 1 (as in fault tolerant network the best forwarding node should be this distance from the destination), $U(0,1)$ is a real random number uniformly distributed between 0 and 1 (randomizing delays to reduce collisions) and λ is a scaling factor that defines the stretch of random delay values.

Equation 3.1 ensures that the nodes closest to the destination have the highest probability of forwarding a packet. If a node overhears another node forwarding the same packet which it is waiting to transmit, it will cancel its own transmission. Upon hearing the packet being transmitted, the sender will also send an acknowledgment (ACK) packet signaling all nodes within its communication range to cancel their transmissions, just in case the self-selected node's transmission is out of range of receivers competing to forward that packet. This process repeats until a packet reaches its destination.

SSR's benefits lie in its low overhead (SSR does not require explicit route maintenance or node location information) and fault-tolerance, since packets are received over all links of the sender and therefore have a high probability of reaching the best available neighbor in each transmission. However, SSR suffers from two limitations.

First, delays based on Equation 3.1 result in packets unnecessarily traveling longer routes even if shorter routes are available. If there are no failures in the network, then it is clear from the way the hop count to the destination is established that each node has at least one neighbor that is one hop closer to the destination than itself. It is also clear that all neighbors must have their hop distances within a

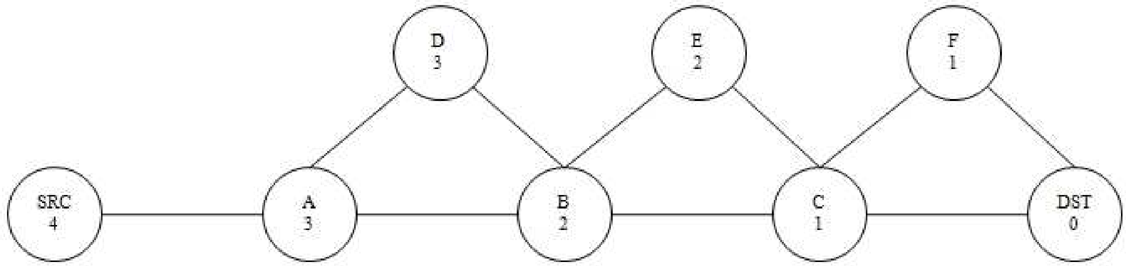

Figure 3.1: Diagram for a packet routing illustration.

small range of the sender. Namely their distances must be at most by one smaller and at most by one greater than its hop distance. The delays generated according to Equation 3.1 may result in a neighbor that is farther from the destination than the sender forwarding the sender's packet, therefore routing a packet via a path longer than necessary. For example, consider the network shown in Figure 3.1, where nodes are represented by circles and their hop distances from the destination (labeled DST) are indicated by the numbers in the circles. Suppose that node A has forwarded a packet from the source (labeled SRC) with an expected hop distance of 2, and node B and D *compete* for forwarding it (node SRC will not try to forward the packet since it just sent it). From Equation 3.1, node B's delay will be $d_{B\_back-off} = \lambda \cdot U(0,1)$ and node D's delay will be $d_{D\_back-off} = 2\lambda \cdot U(0,1)$. The probability that node D will choose to forward the packet is then:

$$p = \int_0^{\lambda} \frac{\lambda - x}{\lambda} \frac{dx}{2\lambda} = \frac{1}{4} \tag{3.2}$$

Therefore, node A's packet has a one in four chance of following a route of length 5 instead of 4. The probability of selecting the longer route of course increases if there are more nodes in the sender's neighborhood through which such a route could be traversed. Hence, Equation 3.1 can be improved to reduce such probability p and therefore enable better performance.

The second limitation of SSR is that it does not support any route repair routine for propagating packets around severed routes, which occur when, for a particular node, all its available neighbors have higher hop distances to the destination than itself. Currently, upon encountering a severed route, a packet may by chance travel backwards towards its source until a new route is found in a way similar to the scenario in Figure 3.1. Relying on such backward travel is inefficient. First, probability of subsequent backward hops drops exponentially with the number of hops, so it is very likely that packet will exceed its time-to-live counter before it reaches the destination in such situation. Additionally, SSR will not adapt its behavior in such a way as to prevent further packets from traveling down the severed route to the cut-off point. These shortfalls in SSR prompted the development of Self-Healing Routing (SHR).

3.2 Self Healing Routing (SHR)

The primary difference between SSR and SHR is the implementation of a route repair, i.e. healing routine. First, upon receiving a DATA packet, instead of using Equation 3.1, a node will ignore the packet if its hop distance is larger than the expected hop distance of the packet plus retransmission bit. Otherwise, it will use the following equation to determine the delay before forwarding the packet:

$$d_{back-off} = \frac{\lambda}{h_{expected} - h + 1 + retransmission} U(0,1) \tag{3.3}$$

As the name indicates in Equation 3.3, retransmission is 0 for the regular DATA packets or packets sent in the route repair step and 1 for packets retransmitted during the resending stage (described later). As in the case of Equation 3.1, delays computed according to Equation 3.3 ensure that those nodes that are closer to the destination than the sender forward their packets before those that are not. Additionally, Equation 3.3 generates delays for nodes that are no closer to the destination than the sender only if there are no responses from the nodes that are closer. Hence, no packet will travel a route longer than necessary.

The second improvement is the addition of a route repair routine for propagating packets around severed routes. As previously mentioned, a severed route occurs when a sending node has neighbors that are all farther from the destination than itself. In this case, corrective action must be taken to reroute packets along the remaining shortest route.

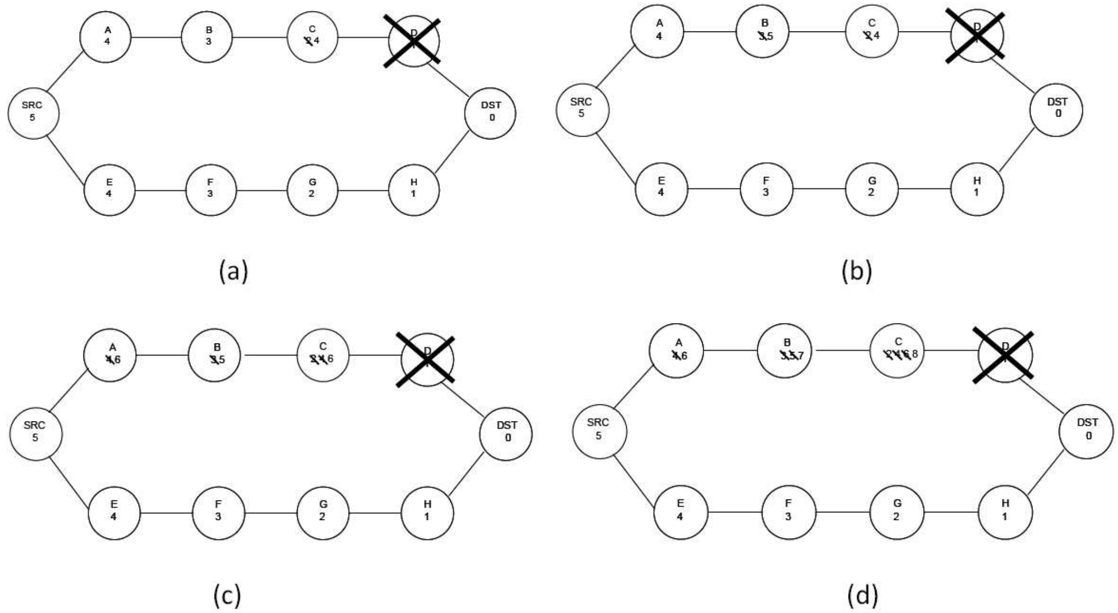

Figure 3.2: SHR Route Repair Scenario

The route repair routine is established so that a node will attempt to forward the packet two times. If at that point it fails to do so, a packet is sent with the hop count to the destination increased by two and the node's stored hop count for the flow is increased by two. This has two effects. The first is an attempt to reroute the packet locally. The second is to prevent the node from winning future competitions to forward a packet along the affected flow.

An example of the route repair routine is given in Figure 3.2, which shows how the route repair scheme works to quickly fix the blocked route. Suppose that node D is either asleep or down and node C has a packet to transmit as shown in Figure 3.2(a). Lack of response to node C's second transmission will cause node C's hop distance to increase to 4 as shown in Figure 3.2(a). When the next packet of the same flow is received by node B, its transmission and retransmission will not have responders; so node B will increase its hop distance to 5 as shown in Figure 3.2(b). The packet then will transmit to node C and it will again transmit and retransmit unsuccessfully, so node C will increase its hop distance to 6 as shown in Figure 3.2(c). The next packet received by node A will not be able to transmit, so node A will increase its hop distance to 6, and trigger transmission of the packet to nodes B and C, increasing their distances to 7 and 8, respectively (see Figure 3.2(d)). In this scenario, the next packet from the source will find the only alternative route via nodes E, F, G, and H, completing the route repair and sending this packet on the route to the destination. From this point on, all packets will travel along the new path.

Although the route repair was initially reported in [11], its costs or even convergence was not established. In [34], an upper bound was established on the cost of route repair in SHR. As already described, in SHR the sender of a packet listens to the response to its transmission. If such a response does not arrive within the time λ, signaling the failure of the previously existing link, the node retransmits the original packet. After the predefined number of unsuccessful retransmissions (two in the current implementation), the sender increases its distance to the destination by 2, as lack of responses to the transmission and retransmissions demonstrates that the only surviving neighbors are nodes with hop distance at least one larger than the current hop distance of the sender. We call such a step a recalibration of the hop distance. Let's consider a sensor network of n nodes in which there is a failure of nodes or their links after which the shortest path from the source to the destination surviving the failure is of length $l < n$. That means that once all nodes not on any of the surviving paths recalibrate their distance to at most n, and the nodes on the surviving paths recalibrate to their correct value, also at most n, then all traffic will flow through the shortest surviving path. The smallest initial distance that nodes needing recalibration might have is 1, so at most $(n-1) * \frac{n}{2}$, hence $O(n^2)$ recalibration steps are needed.

4. Self Selecting Reliable Path (SRP)

4.1 SRP Background

Self Selecting Reliable Path (SRP) was a collaborative effort. The main contributions to the SRP protocol of this thesis are the analytical analysis that a reliable path would always be found in a network, that there was a bound for finding that reliable path, and the comparisons showing how SRP outperformed GRAB, a similar WSN protocol [16]. The author also conducted numerous simulations and proposed the idea for modification to the route repair routine that lead to SRPv2.

As mentioned in the introduction, a WSN protocol must maximize bandwidth use by minimizing end-to-end delay. While SSR and SHR did a good job at forwarding packets there was still a large failure rate and it was considerably slower, in many instances, than protocols that followed the wired network model of having a routing table such as AODV [6]. Part of the issue with SSR and SHR is that each time a packet was sent a new route would often be used. This is ideal if the old route is blocked, but not in a relatively stable network. This led to the, biologically inspired, solution of a reliable path; which was originally introduced in [15] and further discussed in [25]. It was based on the way ants leave a pheromone trail to mark a successful path to a food source or to transfer information about the route to a colony as described in [35].

A scheme to promote a reliable path was introduced in [15]. This preferred path was intended to allow nodes that successfully forwarded a packet to reduce their back-off delay for transmission along the same flow. If a node won at a given hop count it would recalculate its back-off delay by dividing it by 625, while ensuring that the delay was larger than the radio transmission time to avoid collisions. This results in a back-off delay between 20 and 160μs, given λ is 100ms. This reduction in the back-off delay almost guarantees that nodes future selection and stabilizes a path. When a node fails, or there is a transient link, a new node takes its place along the preferred path.

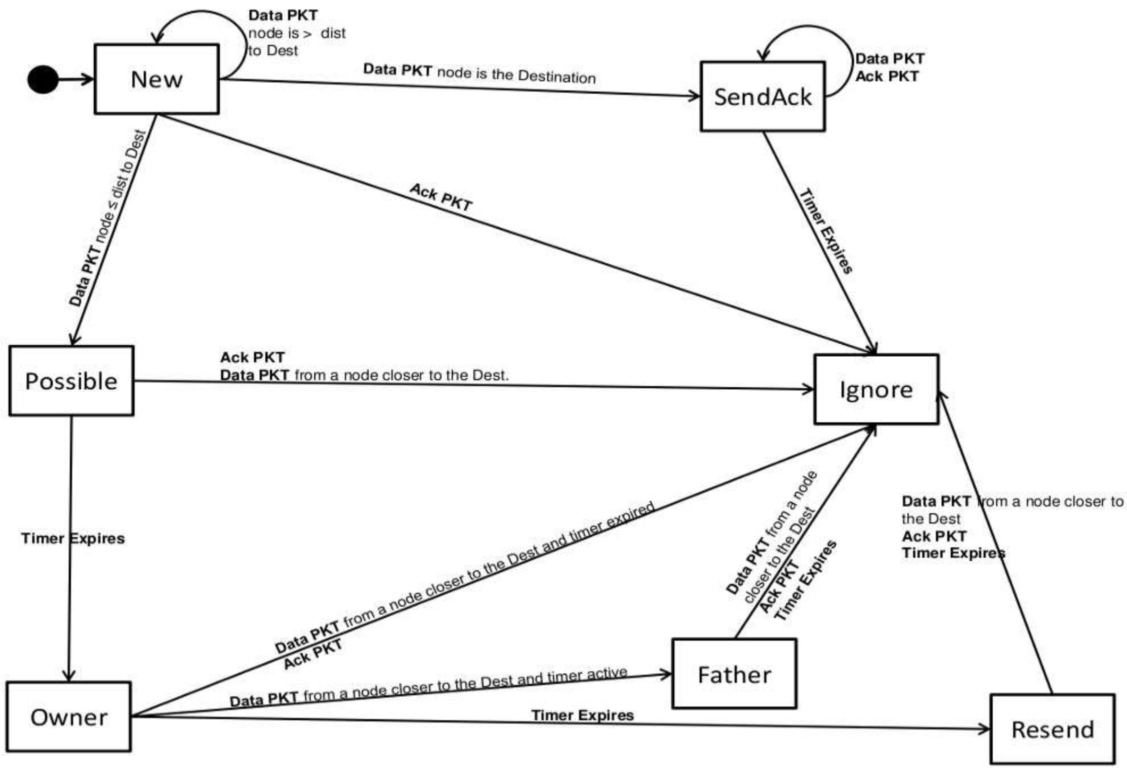

Figure 4.1: State diagram for SRP

4.1.1 SRP Finite State Autonama

Other than path preference, much of the SRP protocol remains the same as SHR. As shown in Figure 4.1, the data transmission stage can be represented by a Finite State Automaton (FSA). This helps to define the input, actions and output generated in each state of a node in the network as it routes data. For example, when a node receives a packet that it has not seen before, it immediately moves into the *NEW* state. It then moves to the correct state depending on the input and status of the node. Different reactions occur if the node is the destination, a node closer to the destination or farther from the destination. The FSA helped in debugging the protocol by enabling visulaization of what occurs at each node.

As seen in Figure 4.1, when the source transmits a DATA packet, only neighbors that are closer to the destination will start at timer. Depending on the proximity to the destination in relation to the sending node, the node selects a transmission back-off delay; this delay is uniformly distributed between 0 and $\frac{\lambda}{2}$ when one hop closer to the destination. If the node is more than one hop closer, there is a high probability of a transient link so the back-off delay is uniformly distributed between $\frac{3\lambda}{4}$ and λ.

λ is a scaling factor that allows for the protocol to tune the probability of collision of the nodes' responses. If, during the back-off delay, a DATA packet is received from a node that is closer to the destination, then the receiving node cancels the forwarding of the DATA packet and moves to the *IGNORE* state. Only when the transmission back-off time expires does the node increment the packet's actual hop count by one, reset the expected hop count to its hop distance to the destination and transmit the packet. Once the the node forwards the packet, it monitors the carrier to determine if the packet was forwarded. If the packet is not forwarded, then the packet is transmitted again. This triggers the route repair routine which was mentioned in the chapter on SHR and will be anylized further in subseqent chapters to justify the need for the improvements made in the RPSP protocol.

4.1.2 Analytical Proof That an Alternate Route will be Found in SRP

This was originally presented in [16] as a justification for why SRP is a viable solution for WSN data transmission protocols. The interesting behavior of SRP arises from the way it selects its routes. If there exists a path from the source to destination on which no transient failures occur, the protocol will converge its routing to such a reliable path. Even more, it will converge to the shortest reliable path. Here is the proof.

Let us consider first a single hop on the currently used path and let $m_s \geq 1$ denote the number of possible forwarders for this hop with stable links to the current sender, while $m_t \geq 0$ denote such forwarders with transient links. Hence, there is a probability $p_s = \frac{m_s}{m_s+m_t}$ that the selected node will have a stable link. Since there is non-zero probability that a forwarding node with transient link will fail to forward and therefore force new self-selection in which nodes with stable links have non-zero probability to succeed, it is clear that in a stable solution, reliable links will be used. To compute the average number of packets needed to get the stable node selected, we have the following:

$$c_{ave} = p_s \sum_{i=1}^{\infty} i\,(1-p_s)^{i-1} = \sum_{i=1}^{\infty} (1-p_s)^{i-1} = \frac{1}{p_s} = 1 + \frac{m_t}{m_s} \tag{4.1}$$

As shown in Equation 4.1, if there is a stable path at all, through route repair, it will be selected after a finite number of packets flow through; even if a path with transient links were selected initially, there is a non-zero probability that all the possible forwarders fail to respond twice in a row, initiating a route repair, resulting in forcing the flow through the shortest stable existing path.

4.2 SRP Performance Evaluation in SENSE

4.2.1 SRP Compared to AODV and SHR

As originally presented in [15] and further analyzed in [16], a large scale network was simulated to compare the performance of SRP, SHR and AODV [6]. AODV is representative of traditional route-based routing protocols which finds the single best route to the destination, stores it in the source or over the route, and uses flooding to repair this route when it becomes damaged. It is also typical in its use of acknowledgments to ensure high delivery ratio at the cost of additional packets sent and received during transmission.

The base configuration for the simulations consists of an 8 unit by 8 unit terrain populated with 500 nodes, each with a nominal transmission range of 1 unit. Simulations use the free space propagation model [36]. The simulated application sends packets of a mean size of 1000 bytes at a mean interval of 40 seconds. In each of the several simulations run, we tested the protocols' performance against a change in one of the following test parameters: (1) the rate of permanent node failures; (2) the rate of transient node failures; and (3) the number of sources communicating with a single destination (base station). SRP and SHR used $\lambda = 100$ms and the maximum hop count equal to the distance to the destination plus $\log 2$ of this distance. We gathered the communication delay at the destination, the packet delivery ratio at the destination and the total number of MAC layer packets transmitted.

In order to determine the success of SRP, three simulations tests were conducted. The first is a Single Destination or sink test. The second two were node or link failure tests; which consisted of testing permanent failures and transient failures. As will be described in the two sections below, SRP performed well.

4.2.1.1 Single Destination Simulations

The first test shows the impact of increasing the number of sources communicating with a single destination; a situation that is common in wireless sensor networks. The results of this test are shown in Figure 4.2. Increased traffic causes more random collisions in SHR, decreasing the delivery ratio. AODV maintained a higher delivery ratio at the cost of an increased number of MAC packets produced and larger communication delay. When the number of sources passes 40, AODV must spend so much time maintaining its topology that its performance drops drastically. SRP on the other hand, maintains an extremely high delivery ratio at very

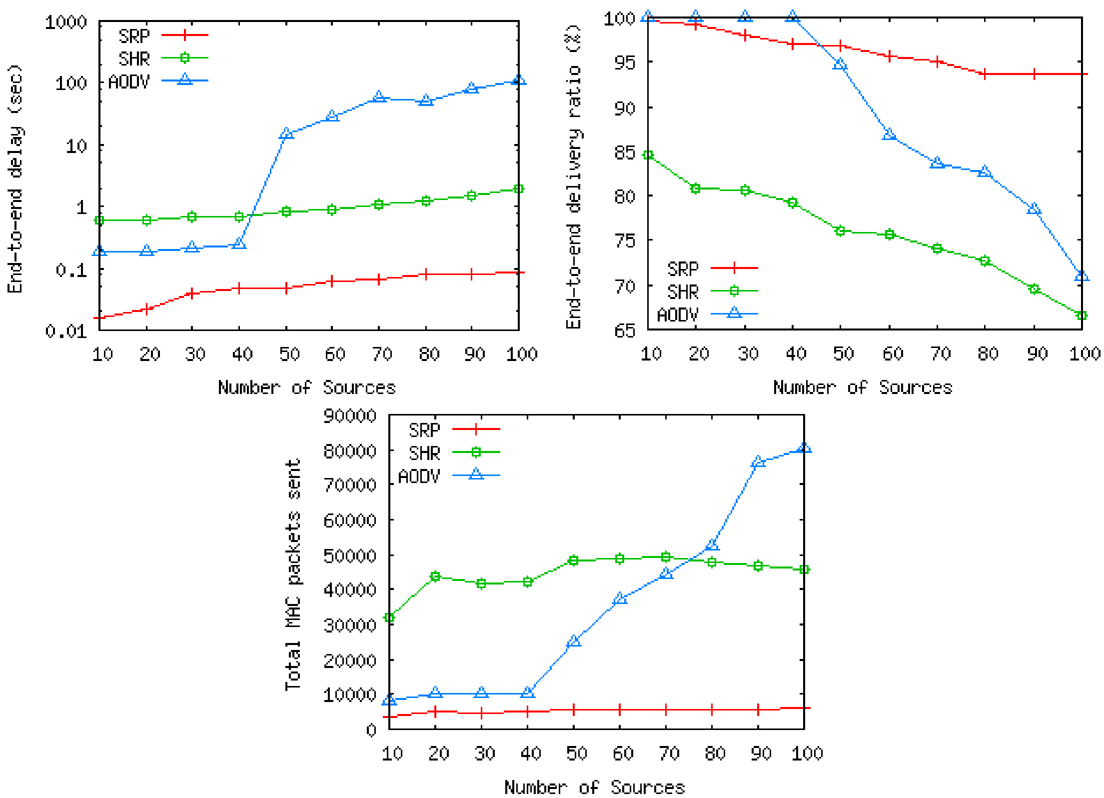

Figure 4.2: Performance of SRP and SHR versus AODV over a reliable sensor network with increasing number of sources reporting to the single base station

quick speeds despite the large increase in traffic. The huge difference in performance between the SSR family protocols and AODV required the use of a logarithmic scale on the end-to-end delay chart. It is also worth noting that, since SRP uses so many fewer MAC packets than AODV, power savings becomes an added, although unintended, benefit.

4.2.1.2 Node Failure Simulations

The next two tests deal with node failure modes. The first to be discussed is permanent failures (see Figure 4.3), followed by transient failures (see Figure 4.4). In sensor networks, transient failures are caused mainly by error-prone links, power management induced duty cycles, and packet collisions. Of these, the duty cycle induced failures are the least disruptive since they are often coordinated with the networking protocol, although this is not the case here. The simulation results

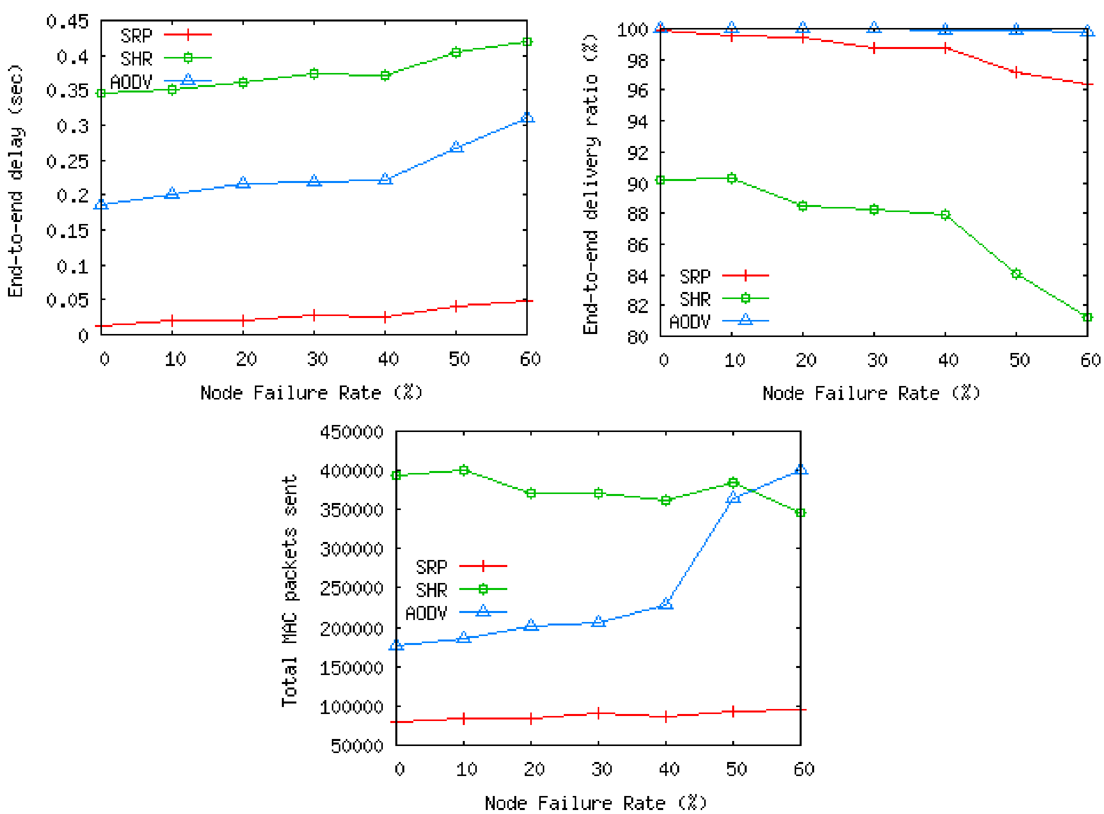

Figure 4.3: Performance of SRP and SHR versus AODV over a sensor network with permanent failures

presented here are based on a random transient failure model, so they exaggerate the effect of duty cycles on the protocols.

When the topology changes, either by a node failing or returning to the network, extra work is required of the networking protocol. The goal is to minimize this work when the failure is transient, yet quickly update the route when the failure is permanent.

When a single permanent failure was introduced at a fraction of the nodes, both AODV and SRP coped well with the disruption and relatively quickly and efficiently found an alternate route. SRP achieved this with smaller delay and significantly fewer packets than AODV, however with a slightly lower delivery ratio as is seen in Figure 4.3.

In case of transient failures, shown in Figure 4.4, AODV is strongly impacted by topology changes. Link layer failures caused AODV to flood the network looking

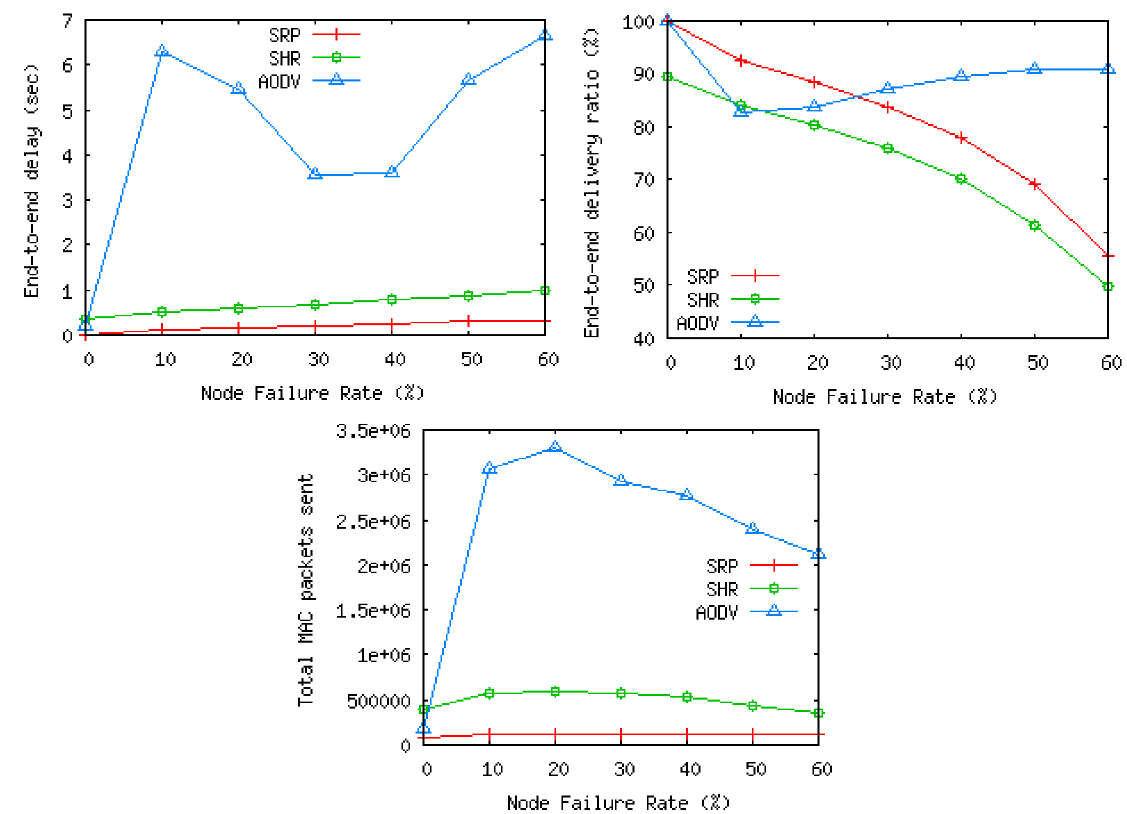

Figure 4.4: Performance of SRP and SHR versus AODV over a sensor network with transient failures

for a new route. The flooding may stop after a few steps, but it is still disruptive. SRP is affected by transient failures (100% delivery rate drops to 57%) but transmits significantly fewer packets than AODV. As the transient failure rate increases, the failures may overcome SRP's ability to repair routes. A simple solution would be to simply send each packet twice in a transient failure prone network, which would increase delivery ratio, maintain faster speeds than AODV, and still use significantly fewer MAC packets.

4.2.2 SRP Compared to GRAB

As part of the research presented in [16], we conducted a number of simulations to compare SRP to the published results of GRAB in [33]. These simulations were conducted to show that SRP could compete against another protocol developed specifically for sensor networks. These dynamic protocols use a similar technique in which nodes compete for forwarding the packet at each hop on the way from the source to destination. The design of GRAB is described in [14]. Using SENSE, we conducted a series of simulations to mimic the ones published in [14], which included delivery rate of the protocol as a function of node failure rate and packet loss rate, as well as delivery rate as a function of network density (total number of nodes in the simulated area).

The authors used a 150 meter by 150 meter topology with 1200 nodes uniformly distributed. They simulated a network with one sink and one source node. The source generated a packet every 10 seconds and sent a total of 100 packets. The nodes were an abstraction of the Berkeley motes [5], which consist of an RF Monolithics 916.50 MHz, transceiver (TR1000) radio that broadcasts with 19.2 Kbps of bandwidth. The transmission and receiving time for a packet was 10ms and the transmitting radius of the radio was 10 meters. Both the two ray and free space signal propagation methods were used but only the two ray results were published. There is a footnote that states that the free space signal model gave similar results. The reported results were averaged over 10 simulation runs.

To match the settings under which those results were obtained, we simulated performance of SRP under both the density test and the permanent failure test. 1200 nodes populated a 15 unit by 15 unit terrain, in which each node is stationary, and has a single unit nominal transmission range. Packets were sent every 10 seconds, and simulation ran for 100 packets. Each simulation was executed ten times, each time with a different random number seed. The same 10 seeds were used for all simulation sets. λ was set to 100ms.

For both tests, the authors of [14] used a 15% link failure rate, which they call a packet loss rate, and either changed the permanent failure rate from 0% to 50% in the failure test, or set it constant at 15% for the density test. We used the permanent failure rate functionality of SENSE. To match the experimental measurements collected in [4, 12] for Crossbow MicaZ nodes, we randomly chose 1/6 of the links as unreliable and dropped 90% of the packets that used those links. This amounts to a total of 15% as the link failure rate (that is packet loss rate reported in [14]). In selecting the transient links in our simulation, we have not considered physical distance from the sender. In a real deployment, most transient links are at the far edges of the radio transmission range. Yet, there can easily be some links that are closer to the sender if an obstacle reduces the transmission range in a particular direction. By choosing 1/6 of the links to be transient, and dropping 90% of packets they overhear, we effectively lost 15% of the packets at the node level.

4.2.2.1 Varying Network Density

In the density simulation we set the permanent failure and link failure rate to 15%. Similar to the simulations reported in [14], ten simulations were run for each density level from 600 to 1800 nodes in increments of 200 nodes. The results for the density test show that SRP is considerably more effective than GRAB in sparse network topologies, as depicted in Figure 4.5.

For a node density of 600, GRAB had approximately 36% delivery rate while SRP's was 60.6%. SRP continued to outperform GRAB until the network size reached 1,000 nodes. At that point, the delivery rate for both protocols stays above 95%. The reason that SRP performs well in sparse networks is that it does not restrict the position of the nodes used for forwarding, like GRAB does, and therefore will find any available route more readily than GRAB.

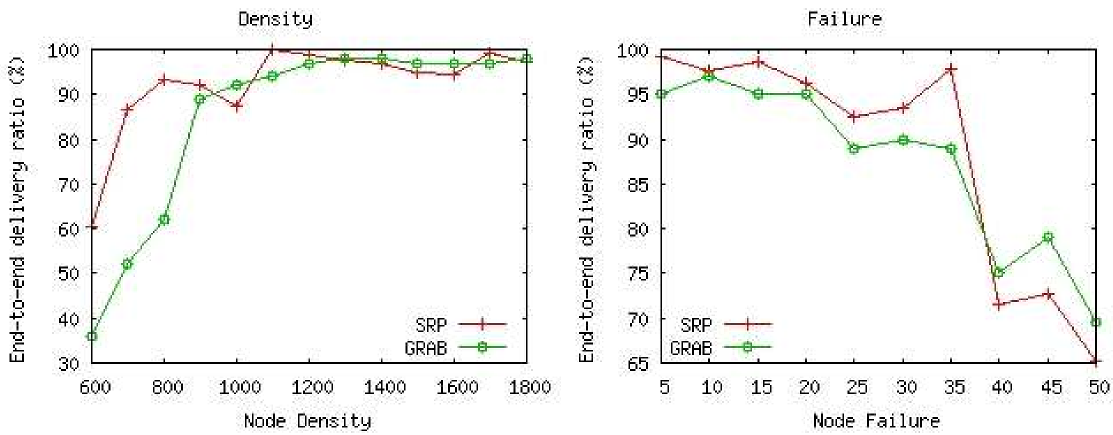

Figure 4.5: Comparison between SRP and GRAB under density and permanent failure tests with a total of 100 packets sent.

4.2.2.2 Varying Network Failure Rate

In the permanent failure simulations the transient link failure rate was set to 15%. Ten simulations were run for each permanent failure rate from 5% to 50% in increments of 5% to get the results comparable to those reported in [14] with the configurations described above. The nodes that failed as part of the permanent failure rate were randomly chosen and failed with probability uniformly distributed over the running time of the simulation. The results for the failure test in Figure 4.5 show that performance of SRP is very comparable to that of GRAB.

At the higher permanent failure rates, GRAB does marginally better. At 50% permanent failure rate, GRAB has approximately 69% delivery rate compared to 65.2% rate achieved by SRP. However at 35% failure rate, SRP's delivery rate of 95% exceeded the 89% of GRAB. Both protocols maintain over 95% delivery rate when permanent failures are less than 20%.

SRP attempts to take advantage of both: (1) dynamic route selection similar to the way GRAB and SSR select paths from source to destination, and (2) static routes that quickly push traffic through a stable network. When the permanent failure rate is 40% or higher, SRP is in complete dynamic selection mode especially when considering those node failures that cause considerable turbulence with a test length of only 100 packets. However, when a semi-stable route can be found, even for a short period of time, the reliable path is quickly established and taken advantage of to speed packets through the network. Existence of such semi-stable routes explains the huge jump in delivery rate for SRP that occurs when the failure rate drops from 40% to 35%. GRAB enjoys a similar jump, but it is not as pronounced. Additionally, when simulations are run longer than for 100 packets the delivery rate of SRP, even with a 50% permanent failure rate, is considerably higher.

5. Reliable Path Self-Selecting Protocol (RPSP)

The introduction of a reliable path in SRP significantly improved the performance of a dynamic route selection protocol in a stable network, as reported in [15, 16]. Yet, there is still the possibility of significant packet loss in the route repair routine for SRP. This led to a new approach to route repair. Two major changes are introduced in RPSP. The first is that a node that forwards a packet returns to a state where it can resend the same packet multiple times, eliminating packet loss that occurs at each iteration of the SRP route repair routine. The second is the addition of a COMP packet.

This chapter is the primary contibution of this thesis and begins with an overview of RPSP and then analytically shows the need for an improvement of the SRP route repair routine. It concludes with simulations that show what environments RPSP and other members of the SSRPF are best suited.

5.1 Overview

Figure 5.1 shows the finite state automata for RPSP. We use the FSA to help express what occurs at the node level and to aid code debugging. In SRP all nodes end at the *IGNORE* state. This was a way to limit multiple paths. All nodes that either won and successfully forwarded a packet or competed and lost ended at the *IGNORE* state. In RPSP, to allow nodes to compete multiple times, nodes go back to the *NEW* state. There is still a need for the *IGNORE* state for any node that had to invoke the repair routine to avoid a packet from getting stuck in an infinite loop. This led to the addition of the *COMP* state that signified that a packet successfully reached the destination.

SRP uses the ACK packet in two ways. First, it stops multiple nodes from forwarding a packet. A node that won self-selection and forwarded a packet is in the *OWNER* state. If that node hears the packet forwarded, it goes to the *FATHER* state. If it hears the same packet forwarded again, signifying a multiple path, an ACK packet is sent to silence all other nodes and the node goes to the *IGNORE*

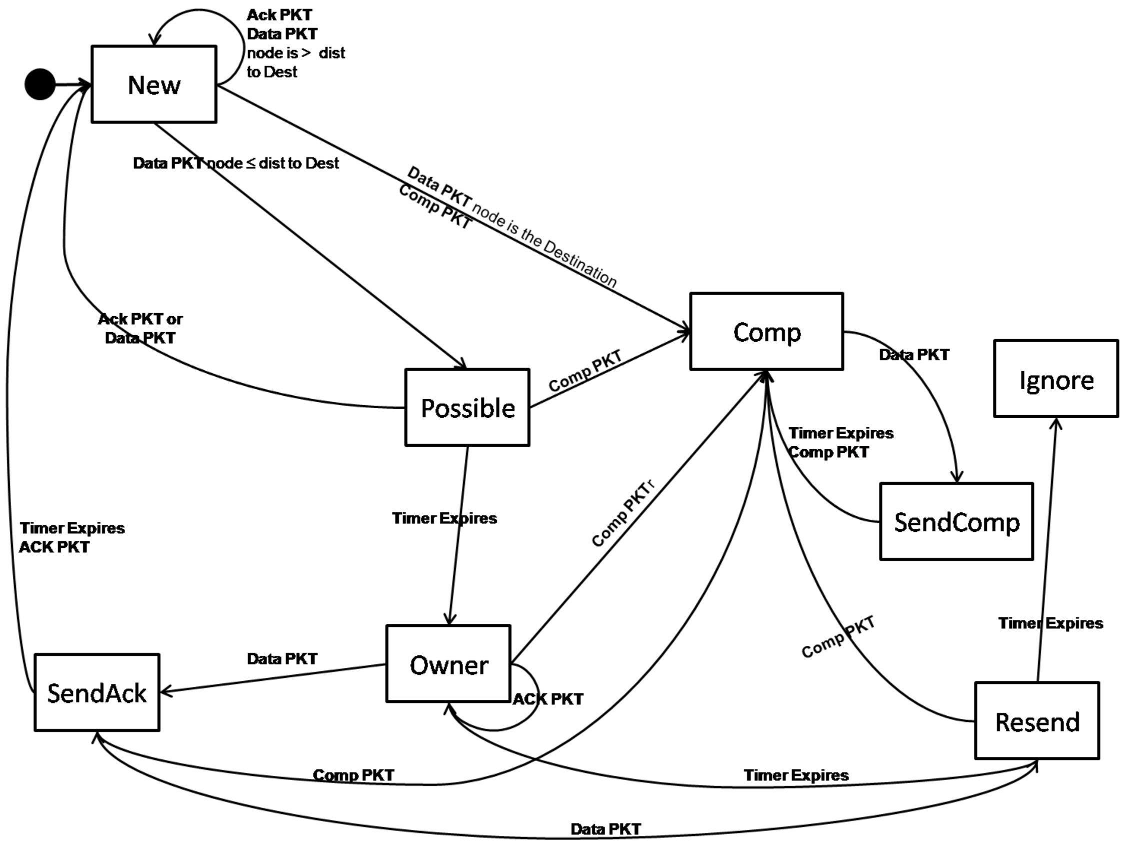

Figure 5.1: RPSP Finite State Automata

state. The second use for the ACK packet is at the destination node which sends it to tell all nodes around it to move to the *IGNORE* state in an attempt to stop multiple paths as far away from the destination as possible. RPSP adds a COMP packet type; it is only used around the destination and retains a similar function to the latter use of the ACK packet in SRP. By adding this packet type, the ACK packet can be used exclusively to silence multiple paths in the network. Looking at Figure 5.1, a winner, in the *OWNER* state, sends an ACK packet immediately upon hearing that the packet is forwarded. This silences all nodes except the next node in the flow. Doing so dramatically reduces any additional paths.

5.2 RPSP Route Repair Routine

This section first demonstrates the inherent problems in SRPv1 and SRPv2. It then shows analytically how the RPSP Route repair routine improves on the

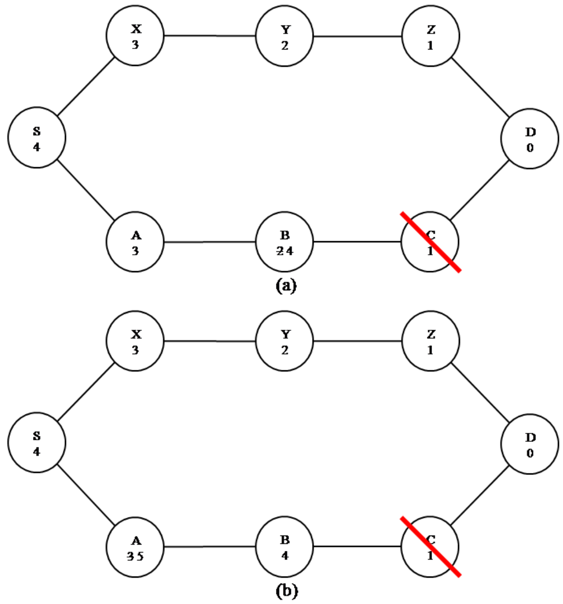

Figure 5.2: SHR/SRPv1 Route Repair Routine

inherent problems in SRPv1 and SRPv2. Finally, it shows test results from SRPv1, SRPv2, ADOV, and RPSP.

5.2.1 Inherent Problems with SRP Route Repair Routines

Both the route repair routine for SRPv1 and SRPv2 work in most situations, but as seen in [16] there are still packets lost during the route repair routine. Figure 5.2 shows the SRPv1 route repair routine and the potential for packet loss. In it, packets flow from the source S to destination D along a reliable path S → A → B → C → D. Then, node C goes down because of a transient link or part of a sleep cycle and the next packet flowing S → A → B encounters an inactive node C (see Figure 5.2(a)), this will cause node B to both increase its hop count to the destination and resend the packet with a hop count of 4. In the state transition, once node A confirms that node B forwarded the packet, it subsequently ignores all additional packets with the same sequence number, resulting in that packet being lost. The following packet will flow S → A (see Figure 5.2(b)), and cause node A to both send the packet with a higher hop count and update the hop count value of node A. This causes a second packet loss. At this point the network is corrected and the next packet will flow S → X → Y → Z → D, which will become the reliable

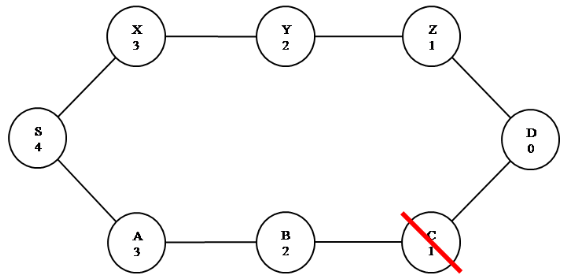

Figure 5.3: SRPv2 Route Repair Routine

path. If following that successful packet transmission, node Z goes down and node C comes back up, then there will be additional packets lost repairing the network again. We will leave it to the reader to go through all the changes in the Figure 5.2, but this process can repeat multiple times or there could be a longer double line scenario, causing significant packet loss.

Figure 5.3 shows SRPv2 route repair routine and its potential for packet loss. Here, packets flow from S $\rightarrow$ D along a reliable path S $\rightarrow$ A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D. If node C fails, then upon receiving a packet, node B will attempt to forward the packet twice and then add two to the expected hop count of the packet header and send the packet a third time maintaining its hop count to the destination. Node A is in the *IGNORE* state resulting in a lost packet. The next packet will follow the same path S $\rightarrow$ A $\rightarrow$ B, again resulting in a lost packet. This will continue until node X wins and forwards the packet. In SHR [11], prior to the idea of a preferred path, each packet send would have a 50% chance for node A or node X to win and forward the packet. In SRP, Node A has a significantly higher chance of winning, as per the backoff delay scheme stated above. Node As backoff delay is $\frac{\lambda}{625}$ while node Xs is a random number between 0 and $\frac{\lambda}{4}$. The average number of packets needed to correct the path would be $\frac{625}{4}$ or approximately 156 packets. This illustrated two key points. The first is that in SRP, the route will correct and forward data. The second is that in some remote situations that could result in a significant number of lost packets.

5.2.2 RPSP Route Repair Routine

In Figure 5.3, RPSP has a reliable path from source S to destination D of S $\rightarrow$ A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D. If node C fails, then node B will attempt to send the packet twice and then, on the third attempt, it will forward the packet with an updated header having an expected hop count of 4, its hop count to the destination plus 2, and go to the *IGNORE* state to avoid a potential infinite loop. In RPSP, node A goes back to the *NEW* state; it will receive the packet and compete for the packet sent by node B. Node S will do the same, as node B and the packet will then follow the alternate path of X $\rightarrow$ Y $\rightarrow$ Z $\rightarrow$ D. This makes the path to the destination S $\rightarrow$ A $\rightarrow$ B $\rightarrow$ A $\rightarrow$ S $\rightarrow$ X $\rightarrow$ Y $\rightarrow$ Z $\rightarrow$ D.

The RPSP route repair routine appears to add both broadcasts and delay to get the packet from source to destination. Consider a n node network arranged into two lines, with a source, a destination and $\frac{n}{2} - 1$ nodes on each line. Additionally, along one line there is a reliable path and its final node prior to the destination fails, as shown to Figure 5.3 for $n = 8$. In SRPv1, SRPv2, and RPSP route repair routines a packet will flow along the reliable path with $\frac{n}{2} - 1$ broadcasts (add one in S and subtract one for the last node). At that point, the route repair routines are called. SRPv1 will lose $\frac{n}{2} - 1$ packets. The final packet lost will broadcast 4 times, all $\frac{n}{2} - 1$ nodes will send $(\frac{n}{2} - 1)(\frac{n}{4} + 3)$ packets in a sequence starting at 4 and adding one recursively for each subsequent node. SRPv2, as shown above, loses on average 156 packets and has $156(\frac{n}{2} - 1)$ or approximately $78n - 156$ broadcasts between successful data transmissions. RPSP will lose zero packets and will have $n - 1$ nodes broadcast (all except the destination), of which $\frac{n}{2} - 2$ nodes broadcasts three times and the rest just once to correct the flow for a total of $2n - 5$ total broadcasts. So, the improved route repair routine for RPSP will both send fewer broadcasts and have fewer packets lost.

5.2.3 Simulations Results Showing the improvement of RPSP

While the weather and physical terrain affect how individual nodes perform and have an impact on the network, they are factors that are constant for a given area. While they will affect performance of the network, they are not instrumental

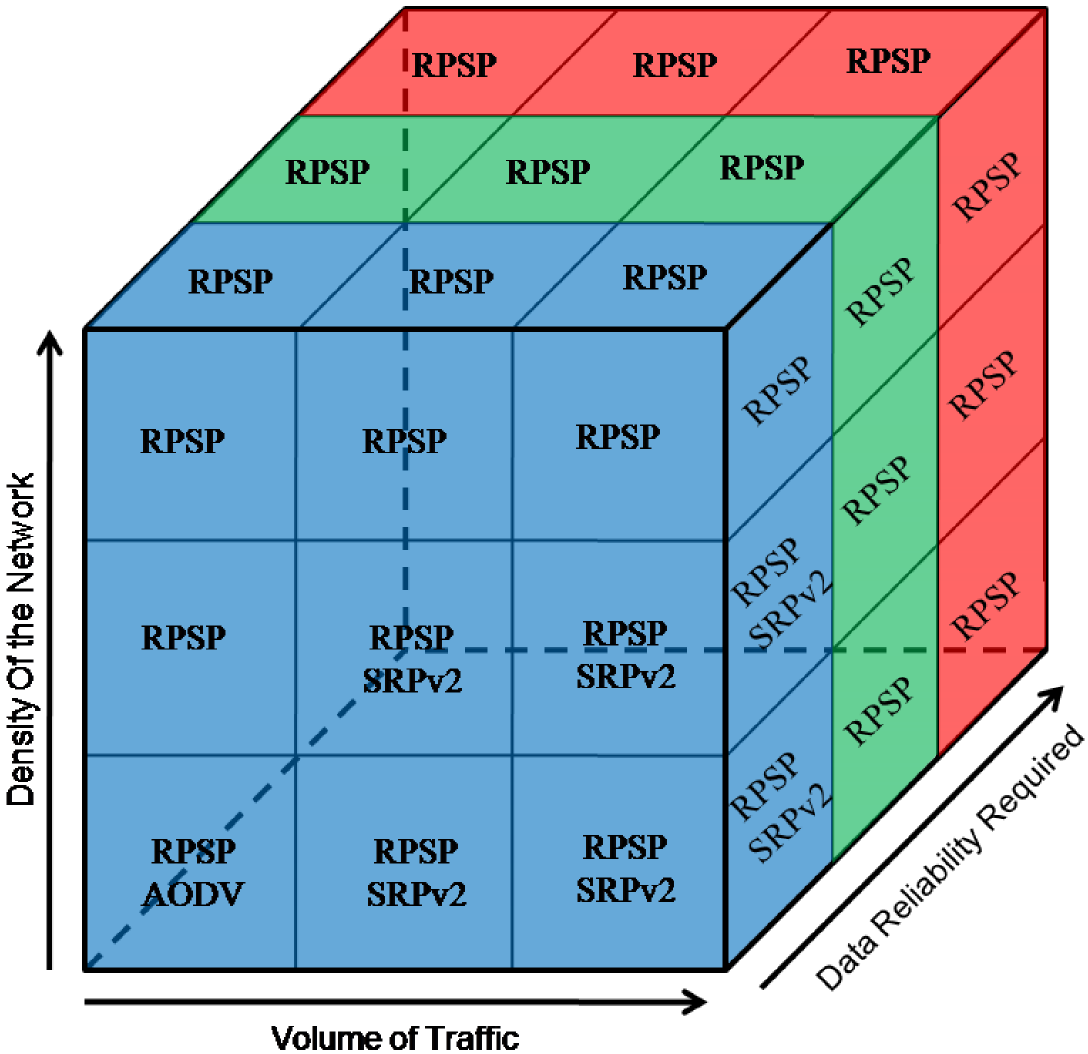

Figure 5.4: Best Suited Protocol

in picking a protocol. There are three major network factors that are controlled by the WSN user: the number of nodes used over a given area (density); the expected frequency of transmissions (bandwidth); and the required data reliability of the application running. We conducted a series of tests to find the best protocol in our suite for the expected use of the WSN. Figure 5.4 shows a diagram of the different considerations. Each block contains the protocol best suited for use given the expected density, network traffic, and data reliability. The subsections discuss the specifics of the results.

To determine the best protocol for use in each environmental condition, we conducted a series of simulations using the SENSE simulator [26]. We conducted two basic tests. The first is a Sink Test in which one destination receives data from a number of sink nodes ranging from 15 to 75 in increments of fifteen nodes. The second is a DutyCycle test, where a certain percentage of nodes failed randomly distributed over a 200 second period and then came back on line, simulating transient links and nodes. The transient failure rate started a 0% and went to 30% in increments of 5% .

Each simulation was conducted at node densities varying from 250, to 500, and to 750 nodes. The simulations were done on a topology consisting of an 8 x 8 unit terrain populated with uniformly randomly placed nodes. Each node is stationary and has a single unit nominal transmission range. The wireless medium is simulated with the free space propagation model [36], and the radio modeled operation at 914 MHz with 1 Mb/s of bandwidth. Packet sizes were uniformly distributed around a mean of 1000 bytes and were sent at uniformly distributed intervals with a mean of 40 seconds. MAC broadcast was used in which a node senses the carrier and broadcasts only if no other transmissions are detected. Each simulation was executed six times, each time with a different random number seed for a simulation time of 3,000 seconds per seed. Each test set used the same seeds for all simulations. λ was set to 100ms for all simulations.

5.2.3.1 Sink Test

In many WSNs, there are a large number of nodes that send data to a central sink that aggregates data for future use. This use pattern plays a significant role in determining which protocol is best suited for the given node density and end-to-end delay. Figure 5.5 shows the results from the sink test.

While AODV does well with few sources, as the number of sources increases from 45 to 60, its end-to-end delivery ratio goes from almost 100% to 96% for 250 and 500 nodes to 95% for 750 nodes. RPSP maintains over 97% delivery ratio regardless of the node density. As the number of source nodes goes to 75, AODV performs at 94% with a node density of 250 and 500 nodes. When the node density is high, as it is in case of 750 nodes, the delivery ratio drops to 70% .

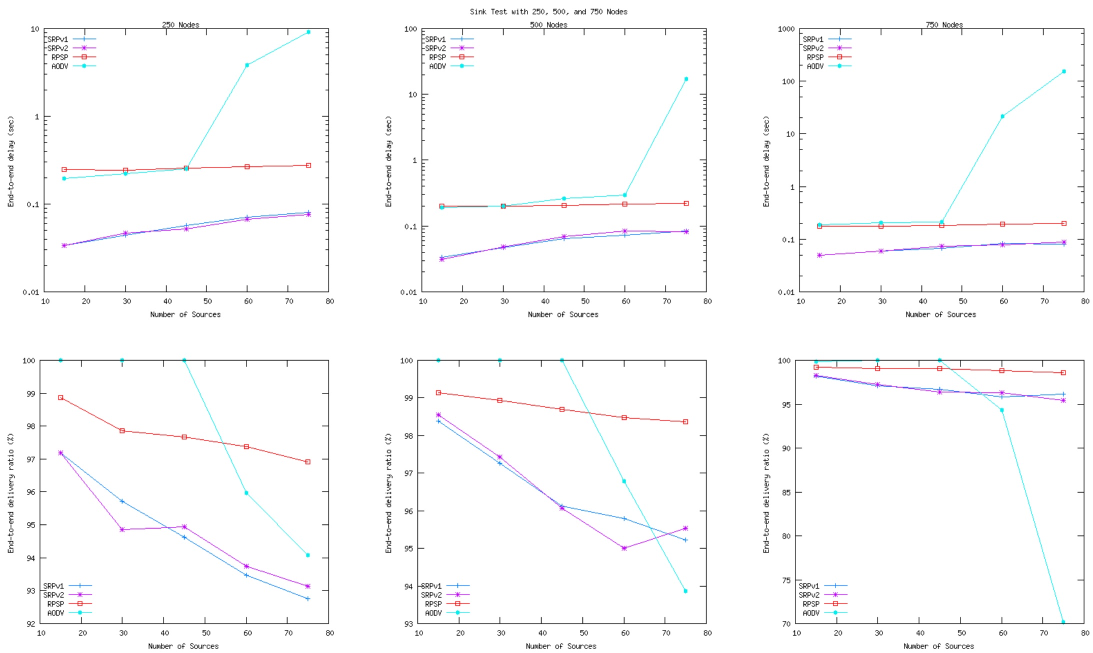

Figure 5.5: RPSP Sink Test

RPSP makes an improvement over SRPv1 and SRPv2 in terms of end-to-end delay, as see in Figure 5.5. It maintains a better end-to-end delay for all node densities.

The end-to-end delay is significantly affected in AODV when the number of sources is increased. RPSP is more likely to stop a reliable path than SRP and has a higher end-to-end delay; however, it remains below 0.5 seconds throughout all of the simulations.

5.2.3.2 Duty Cycle Test

The Duty Cycle test is designed to show how a protocol reacts to transient nodes and links which occur frequently either due to the environment, node failure caused by power exhaustion, or nodes put in sleep mode by an energy saving algorithm. Figure 5.6 shows the results for the duty cycle test. For end-to-end delay, RPSP as expected is higher than SRPv1 and SRPv2. As discussed earlier in the route repair routine section, RPSP should lose fewer packets because there are no packets lost during a successful route repair. RPSP is only slightly better that SRP in low node densities; however it is significantly better in higher node densities than

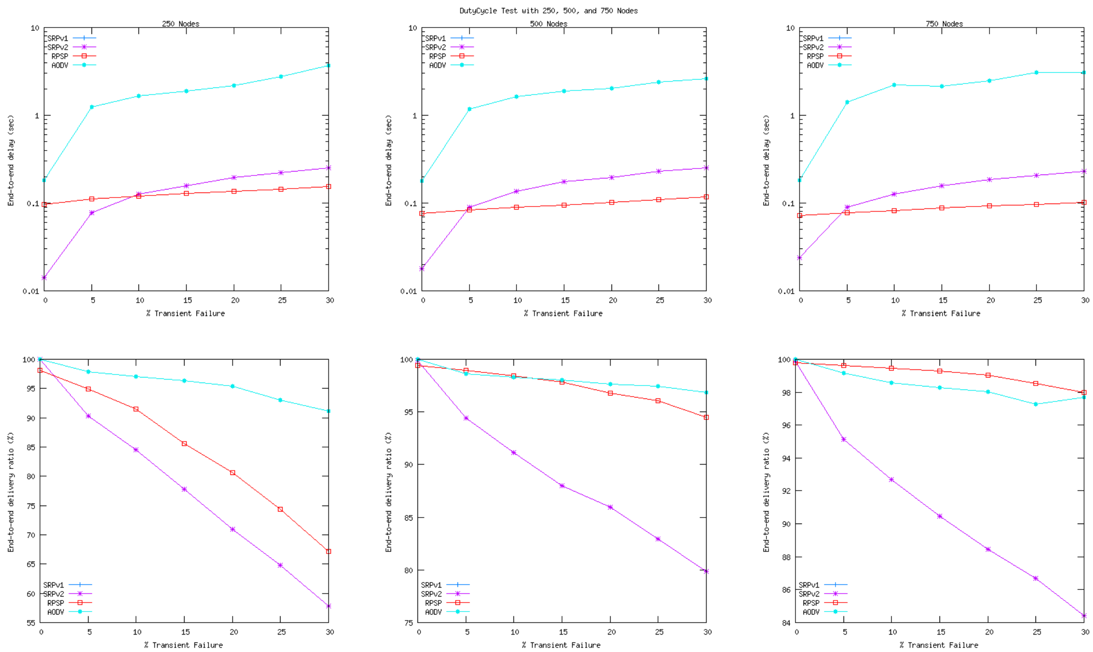

Figure 5.6: RPSP DutyCycle Test

SRP. RPSP additionally maintains roughly that same end-to-end delay no matter what the node density, while SRP has a slight increase in end-to-end delay as the node density increases.

As expected, AODV does better in a less dense network. As the node density increases, AODV has to send considerably more packets to maintain the network as nodes fail. AODV becomes worse as the node density increases to 750 nodes, when there are a large number of transient failures.

6. Discussion and Conclusions

In this thesis, we have introduced RPSP as the newest member of the Self Selecting Routing Protocol Family. Its route repair routine makes it well suited for most operating environments. Additionally, through simulation we have shown that for any operating environment, there is a member of the SSRPF that will perform well. Figure 5.4 above shows the best protocol in the SSRPF for each operating environment base on the simulation results shown in Figure 5.5 and Figure 5.6. Clearly, only in a small part of the overall environment diversity space, namely for medium or high volume of traffic, medium or low density and highly reliable networks, SRPv2 delivers performance comparable to RPSP. Even in a smaller subspace, defined by low volume traffic over highly reliable and low density networks, can AODV rival the performance of RPSP. Only in a few settings, AODV bettered RPSP on deliver ratio metric. Overall, however, RPSP delivers the most reliable fast communication using the fewest number of packets over the majority of the wireless sensor network operating environments.

Future work on SSRPF includes improving the protocols in the family to minimize energy consumption and adapting them to route effectively in environments with mobile nodes. The first extension requires addressing the challenge of limiting overhearing of packet transmission. The second extension needs to address the challenge of efficiently updating the hop distance to the destination. The latter challenge is easier to address when there is a mixture of mobile and stationary nodes in the network, enabling the mobile nodes to learn their hop distances from the stationary ones.

LITERATURE CITED

[1] A. Woo, T. Tong, D. Culler: Taming the underlying challenges of reliable multihop routing in sensor networks. Proc. ACM SenSys03, ACM Press, New York, 2003, pp. 14-27.

[2] J. Zhao, R. Govindan: Understanding packet delivery performance in dense wireless sensor networks. Proc. ACM SenSys 03, ACM Press, New York, 2003, pp. 113.

[3] G. Anastasi, A. Falchi, A. Passarella, M. Conti, E. Gregori: Performance measurements of motes sensor networks. Proc. 7th ACM Intern. Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems, ACM Press, New York, 2004, 174-181.

[4] Crossbow Technology, Inc., http://www.xbow.com

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister: System architecture and directions for networked sensors Proc. 9th ACM Int. Conf. Architectural Support for Programming Languages and Operating Systems:pp. 93-104, 2000.

[6] C. Perkins, E. Belding-Royer, S. Das: RFC 3561-ad hoc on-demand distance vector(AODV) routing. http://www.faqs.org/rfcs/rfc3561.html

[7] C. Intanagonwiwat, R. Govindan, D. Estrin: Directed diffusion: a scalable and robust communication paradigm for sensor networks. Proc. ACM MobiCom, ACM Press, New York, 2000, pp. 56-67.

[8] G. Chen, J.W. Branch, B.K. Szymanski: A Self-Selection Technique for Flooding and Routing in Wireless Ad-Hoc Networks. Journal of Network and System Management, 14(3), 2006, pp. 359-380.

[9] G. Chen, J.W. Branch, B.K. Szymanski: Self-Selective Routing for Wireless Sensor Networks. Proc. of IEEE Int. Conf Wireless and Mobile Computing, Networking, and Communication, WiMob'05, 2005, Vol. 3 ,pp. 57-65.

[10] G. Chen, J.W. Branch, B.K. Szymanski: Local Leader Election, Signal Strength Aware Flooding, and Routeless Routing. Proc. 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc Networks and Sensor Networks, WMAN05, 2005.

[11] J.W. Branch, M. Lisee, B.K. Szymanski: SHR: Self-Healing Routing for wireless ad hoc sensor networks. Proc. Intern. Symp. Performance Evaluation

of Computer and Telecommunication Systems SPECTS'07, SCS Press, San Diego, 2007, pp. 5-14.

[12] K. Wasilewski, J. Branch, M. Lisee, B.K. Szymanski: Self-healing routing: a study in efficiency and resiliency of data delivery in wireless sensor networks. Proc. Conference on unattended Ground, Sea, and Air Sensor Technologies and Applications, SPIE Symposium on Defense & Security, April, Orlando, FL (2007).

[13] R. Poor: Gradient routing in ad hoc networks. http://www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf

[14] F. Ye, G. Zhong, S. Lu, L. Zhang: Gradient broadcast: a robust data delivery protocol for large scale sensor networks. ACM Wireless Networks, 11(2) (2005).

[15] B.K. Szymanski, C. Morrell, S.C. Geyik, T. Babbitt: Biologically Inspired Self Selective Routing with Preferred Path Selection. Bio-Inspired Computing and Communication, LNCS, vol. 5151, Springer, New York, NY, 2008, pp. 217-228.

[16] T. Babbitt, C. Morrell, B.K. Szymanski, J. Branch: Self-Selecting Reliable Path for Wireless Sensor Network Routing. Computer Communication Journal, vol. 31, no. 16, 2008, pp. 3799-3809.

[17] M. Heissenbttel, T. Braun, T. Bernoulli, M. Waelchli: BLR: beaconless routing algorithm for mobile ad hoc networks. Computer Communications Journal, 27(11)(2004).

[18] M. Zori, R.R. Rao: Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. IEEE Trans. Mobile Computing, 2(4) (2003) 337-348.

[19] B. M. Blum, T. He, S. Son, J.A. Stankovic: IGF: a robust state-free communication protocol for sensor networks. Technical Report CS-2003-11, University of Virginia, Charlottesville, 2003.

[20] Y. Xu, W.C. Lee, J. Xu, G. Mitchell: PSGR: priority-based stateless geo-routing in wireless sensor networks. Proc. IEEE Conf. Mobile Ad-hoc and Sensor Systems, IEEE Computer Society Press, Los Alamitos, 2005.

[21] D. Chen, J. Deng, P.K. Varshney: A state-free data delivery protocol for multihop wireless sensor networks. Proc. IEEE Wireless Communications and Networking Conf., IEEE Computer Society Press, Los Alamitos, 2005.

[22] K. Fall, K. Varadhan (eds.): The ns manual (formerly ns notes and documentation). The VINT Project, 2008, http://nsnam.isi.edu/nsnam/index.php.

[23] S. Kurkowski, T. Camp, N. Mushell, M. Colagrosso: A visualization and analysis tool for ns-2 wireless simulations: inspect. Proc. of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) (2005), 503-506.

[24] C. Morrell, T. Babbitt, B.K. Szymanski: Visualization in Sensor Network Simulator, SENSE and Its Use in Protocol Verification. Technical Report 08-13, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, 2008.

[25] C. Morrell: Path Preference In Self-Healing Routing Verified And Improved Through Visualization In Sense. Masters Thesis, Computer Science Department, RPI, 2008.

[26] G. Chen, J. Branch, M.J. Pflug, L. Zhu, B. Szymanski: Sense: A sensor network simulator. Advances in Pervasive Computing and Networking (2004), 249-267.

[27] G. Chen, B.K. Szymanski: Cost: A component-oriented discrete event simulator. Proc. Winter Simulation Conference, WSC02 (San Diego, CA), vol. I, December 2002, pp. 776-780.

[28] B.K. Szymanski, G.G. Chen: Sensor network component based simulator. Handbook of Dynamic System Modeling (Paul Fishwick, ed.), CRC/Taylor and Francis Publishing, 2007, pp. 35-1 – 35-16.

[29] T.T. Huynh, C.S. Hong: An energy delay efficient multi-hop routing scheme for wireless sensor networks. IEICE Transactions on Information and Systems E89(D5) (2006) 6541661.

[30] H.L. Xuan, S. Lee: Two energy-efficient routing algorithms for wireless sensor networks. Networking, LNCS, Springer, New York, NY, 2005, pp. 698-705.

[31] H.K. Ryu, Y.Z. Cho, D.H. Kim, K.W. Lee, H.D. Park: Improved handoff scheme for supporting network mobility in nested mobile networks. Computational Science and Its Applications, LNCS, Springer, New York, NY, 2005, pp. 344-347.

[32] T.T. Huynh, C.S. Hong: A novel hierarchical routing protocol for wireless sensor networks. Mobile Communications Workshop, LNCS, Springer, New York, NY, 2005, pp. 339-347.

[33] G. Chen, J. Branch, B.K. Szymanski: Local leader election, signal strength aware flooding, and routeless routing. 5th IEEE Intern. Workshop Algorithms for Wireless, Mobile, Ad-Hoc Networks and Sensor Networks WMAN 2005. IEEE Computer Society Press, Los Alamitos (2005).

[34] B.K. Szymanski, G. Chen: Computing with Time: From Neural Networks to Sensor Networks. The Computer Journal, vol. 51(4):511-522, 2008.

[35] S. Koenig, B.K. Szymanski, Y. Liu: Efficient and Inefficient Ant Coverage Methods. Annals of Mathematics and Artificial Intelligence 31(1-4), 4176 (2001).

[36] T.S. Rappaport: Wireless Communications: Principles and Practice. Prentice Hall, Englewood Cliffs (1996).

[37] J. Glaser, D. Weber, S.A. Madani, S. Mahlknecht: Power aware simulation framework for wireless sensor networks and nodes. EURASIP Journal on Embedded Systems 2008 (2008).

[38] D. Estrin, M. Handley, J. Heidemann, S. Mccanne, Y. Xu, H. Yu: Network visualization with the VINT network animator nam. Technical Report 99-703, University of Southern California, 1999.

[39] S. Kurkowski, T. Camp, M. Colagrosso: A visualization and analysis tool for wireless simulations: inspect. ACM's Mobile Computing and Communications Review, to appear (2008).

[40] C. Johnson: Visualization viewpoints. IEEE Computer Graphics and Applications (2004), 13-17.

[41] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Halmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu: Advances in network simulation. IEEE Computer 33(4) (2000), 59-67.

[42] C. Goldstein, S. Leisten, K. Stark, A. Tickle: Using a network simulation tool to engage students in active learning enhances their understanding of complex data communications concepts. 7th Australasian conference on Computing education, Australian Computer Society, Darlinghurst, Australia, 2005, pp. 223-228.