Proc. IEEE International Symposium on Computers and Communication, Sousse, Tunisia, July 5-8, 2009, pp. 1-7

Self-Selecting Reliable Path Routing in Diverse Wireless Sensor Network Environments

Thomas Babbitt, Christopher Morrell, and Boleslaw Szymanski
Department of Computer Science, RPI, Troy, NY
{babbit, morrec, szymansk}@cs.rpi.edu

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 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 suffer from a delay required to make such selection. 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 paper 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.

1. Introduction

A Wireless Sensor Network (WSN) consists of numerous sensor nodes that are linked into a wireless network. There are many applications for which WSNs are well suited [1]. A large majority of WSNs require battery powered nodes capable of surviving harsh environments for extended periods of time. Moreover, WSN must be autonomous, fault tolerant, and energy efficient. These requirements are critical in routing, because multi-hop transmission is an extremely fault prone and energy consuming operation. For example, commonly occurring in WSNs are faulty or ill placed nodes and transient links causing an oscillation of packet reception quality which can cause severe packet loss and spontaneous network topology changes [2], [3]. Studies show that most WSN operational energy is used for radio operations [4]. Typical hardware specifications are listed in [5] and [6].

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 the 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 application's 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 [7], MintRoute [8], and Directed Diffusion [9] 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 [10], [11], SHR [12], GRAd [13], and GRAB [14]. The final challenge is to both efficiently use the bandwidth and maintain dataflow while minimizing energy use. 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. A recent review of WSN and energy saving algorithms is given in [15], which also includes protocols for ad-hoc networks.

This paper presents a novel family of wireless sensor routing protocols, the Self-Selecting Reliable Path Routing Protocol Family (SSRPF), that address all three challenges. There are three members of the SSRPF family. The first is Self-Selecting Reliable Path (SRPv1) protocol [16] 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 (SRPv2) protocol [17] 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), introduced in this paper, which modifies the route repair routine to eliminate the lost packets that occur in the repair routine for SRP.

The rest of this paper is organized as follows. Section 2 describes our research background on SRP. The new contributions to protocol design, mainly the improvement of the route repair routine from SRP to RPSP, are described in Section 3. Sections 4 and 5 compare the members of the SSRPF family with AODV, SHR and GRAB in the prevalent operating environments using the SENSE Simulator [18]. Section 6 contains conclusions and outlines future work.

2. SRP Overview

The inspiration for SRP, specifically the addition of the reliable path, came from observations on ant colonies [16]. The reliable path selection closely resembles how ants use pheromones to mark a path from a food source to their colony. When the non-scouting ant goes out, it follows the path with the strongest pheromone levels in an attempt to reinforce success. Our reliable path does the same by allowing nodes to quickly self select if they previously won and forwarded a packet based on a simple back-off delay scheme.

When a data packet is sent from a source to a destination, each node competes for self selection based on the following back-off delay scheme. The node that received a packet with the given distance to the destination, selects its delay depending on the condition it satisfies, as follows.

1. If it is a part of the reliable path, its delay is $\lambda/625$ (enough time to ensure that another node is not transmitting).
2. Otherwise, if it is one hop closer to the destination, its delay is selected randomly from range $(0,\lambda/4)$.
3. Otherwise, if it is more than one hop closer, its delay is $\lambda/4$+ random delay selected from range $(0,\lambda/4)$.
4. Otherwise, if it its distance is equal to packet's distance, its delay is $\lambda/2$+random delay selected from range $(0,\lambda/2)$.
5. Nodes father than the packet's distance ignore the packet.

In the above formulas λ denotes the range over which the response messages are distributed. The probability of collision of two response messages is thus proportional to the number of nodes competing for response and inversely proportional to this range λ. However, the average delay on each hop is proportional to this range, unless the hop is on the preferred path. Thanks to routing most of the packets via the preferred paths in SSRPF family of protocols, the value of range λ can be selected large, so the probability of collision of responses is below 1%. This is a major improvement over the earlier version, were low probability of response message collisions was paid by the delay of forwarding the packet at each hop.

Through experimentation on Micaz nodes [5], we found that the best forwarders are nodes one hop closer to the destination than the sender [12], so the scheme favors such nodes. Nodes more than one hop closer have a considerably higher chance of having a transient link and may not be stable in a reliable path; they are given a separate and lower priority to help avoid them, if possible. In cases where a node wins and there is no subsequent node closer to the destination, a node at the distance equal to the packet's distance can win and ensure that the packet is forwarded on. This is a last resort choice because it adds both additional time and packet transmission, affecting end to end delay, energy use, and potentially delivery ratio. Even with the ability for nodes to self select using the aforementioned back off delay scheme, there can be considerable packet loss in the route repair routines

2.1. SHR and SRP Route Repair Routines

The original route repair routine description was given in [11]. A timer is set once a node forwards a packet. If the node has not heard the packet forwarded and the timer expires, then the packet is sent again. This can be done multiple times. The more times the packet is retransmitted, the more likely a node closer to the destination than the sender will hear and forward it; however, it also slows the end to end transmission of the packet and could waste energy if an active node closer to the destination in the transmission radius does not exist. We attempt to send the packet twice. Once the second timer expires then the node adds two to its expected hop count to the destination in both the packet header and in the node itself. This does two things; it both enables the packet to be forwarded by a node that previously had a higher hop count to the destination and it prevents that node from forwarding packets along a dead end path in the future.

Originally, as described in [11], there could be a hello packet sent when a node came back online. This packet would then propagate its hop count to the destination out to its neighbors allowing them to update their hop counts. The idea of the hello packet was to correct the distances to the destination. While the idea was a good one, it did not work in practice because usually a node does not know when it is reentering a network.

A simple change to the route routine is to only update the packet's header and resend the packet. This simple change helped to keep nodes from changing their hop count and altering the way the network topology looked. A simple example of the network being adjusted is below. This is preferred especially when introducing sleep schedules which could alter reliable paths and cause self induced network updates.

2.2. SRP Route Repair Routine Problems

Both route repair routines work in most situations, but as seen in Fig. 1 (adopted from [17]) there are still packets lost during the route repair routine of SRPv1. As shown in Fig. 1, 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 first packet flowing (S,A,B) and encountering inactive C (see Fig. 1), will cause node B to both increase its hop count to the destination and

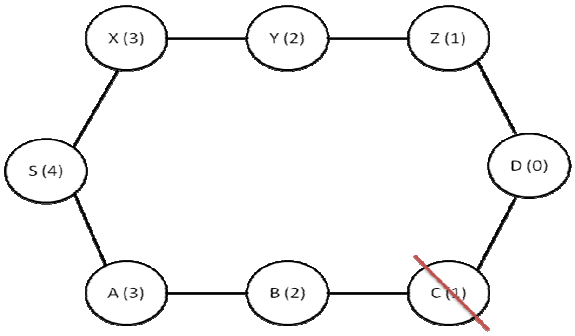

Fig. 1. SHR/SRPv1 Route Repair Routine

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 the loss of this packet. The following packet will flow (S,A), 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 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. This process can repeat multiple times or there could be a longer double line scenario, causing significant packet loss.

In SRPv2 route repair routine, neither node B nor A will change their hops to destination. After node C fails, node B, upon receiving a packet, 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, as sender, will ignore the packet, so it will be lost. The next packet will follow the same path (S,A,B), again resulting in a packet loss. This will continue until node X wins and forwards the packet. In SHR [12], prior to the idea of a preferred path, each packet sent 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 A's backoff delay is $\lambda/625$ while node X's is a random number between 0 and $\lambda/4$. The average number of packets needed to correct the path would be 625/4 or approximately 156 packets. This illustrates two key points. The first is that in SRP, the route will correct and forward data. The second is that in some unlikely situations that could result in a significant number of lost packets.

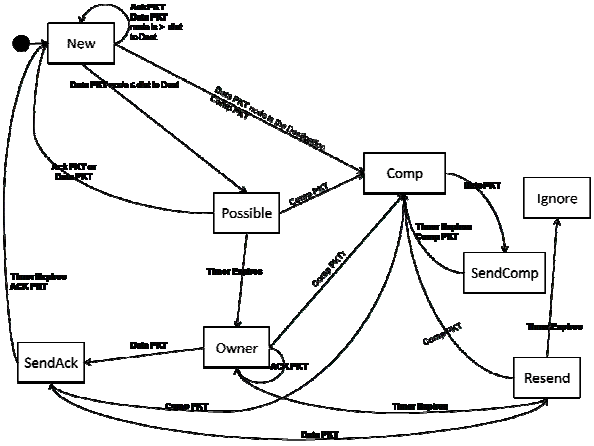

Fig. 2. FSA of RPSP

3. RPSP

The introduction of a reliable path in SRP significantly improved the performance of a dynamic route selection protocol in a stable network [17]. Yet, as seen above, 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, presented here.

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. RPSP maintains the reliable path introduced in SRP.

Fig. 2 shows the finite state automata for RPSP which expresses what occurs at the node level and aids code debugging. In SRP all nodes that either won and successfully forwarded a packet or competed and lost move to the ignore state to limit creation of multiple paths. 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 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 Fig. 3, 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 path. Doing so dramatically reduces any additional paths.

In Fig. 1 above, RPSP has a reliable path from source S to Destination D of (S,A,B,C,D). If node C fails, then node B will attempt to send the packet twice. 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,Y,Z,D). This makes the path to the destination going back to the source, (S,A,B,A,S,X,Y,Z,D).

The RPSP route repair routine appears to add both broadcasts and delay to get the packet from source to destination. Consider n node network arranged into a double line, with a source, a destination and 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 Fig. 1 for n=8. In SRPv1, SRPv2, and RPSP route repair routines a packet will flow along the reliable path with 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 n/2-1 packets. The final packet lost will broadcast 4 times, all n/2-1 nodes will send (n/2-1)(n/4+3) packets in a sequence starting at 4 and adding one recursively for each subsequent node.

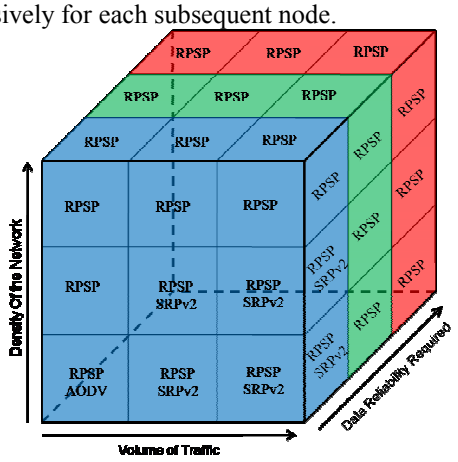

Fig. 3. Best Suited Protocol

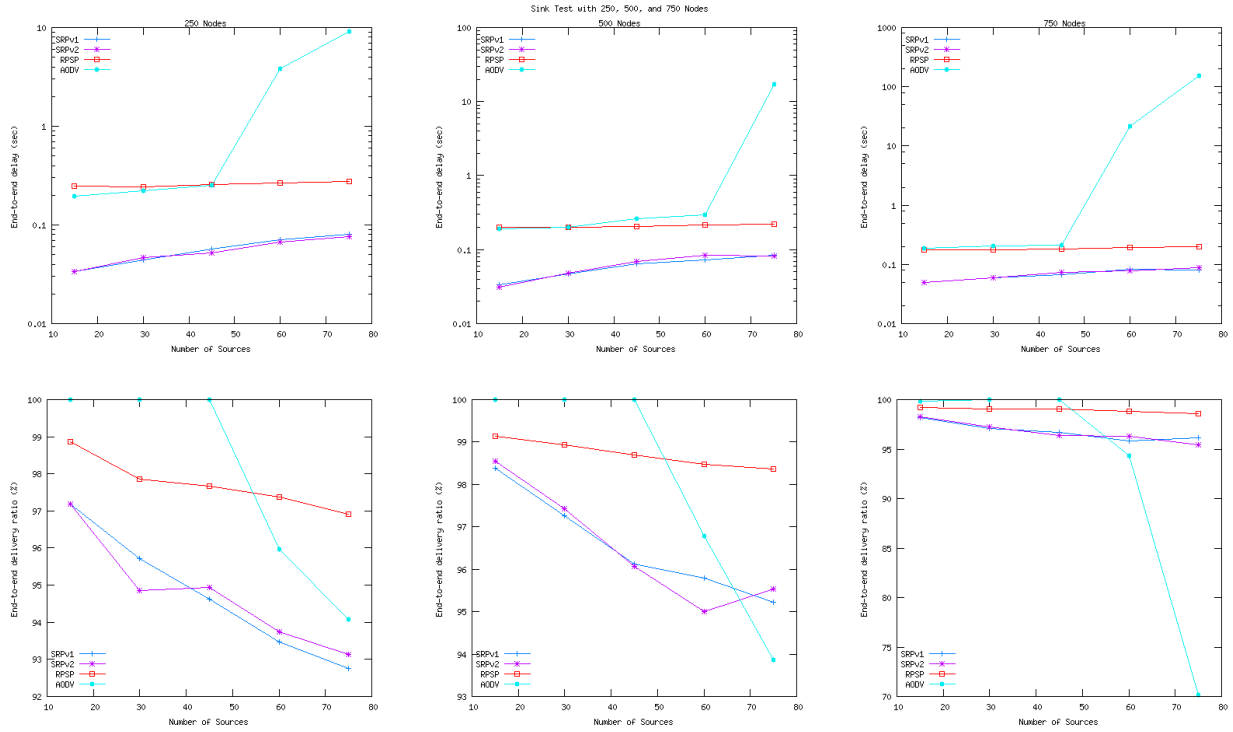

Fig. 4. Sink Test

SRPv2, as shown above, loses on average 156 packets and has 156(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 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.

4. Environmental Conditions

While the weather and physical terrain effect 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 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 data reliability required of the application. We run a series of tests to find the best protocol in our suite for the expected use of the WSN. Fig. 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 simulation section discusses the specifics of the results.

5. Simulations

To determine the best protocol for use in each environmental condition, we conducted a series of simulations using SENSE simulator [18] that is available publicly at www.ita.cs.rpi.edu/sense/. We conducted two basic tests. The first is a Sink Test in which one destination receives data from a number of source nodes ranging from 15 to 75 in increments of fifteen nodes. The second is a DutyCycle test where a certain percentage of nodes fails over randomly distributed 200 sec. period and then came back online, simulating transient links and nodes. The transient failure rate started a 0% and went to 30% in increments of 5%.

The simulations were done on a topology consisting of an 8 x 8 unit terrain populated with uniformly randomly placed nodes. Each node was stationary and had a single unit nominal transmission range. Each simulation was conducted at node densities varying from 250, to 500, and to 750 nodes. The wireless medium was simulated with the free space propagation

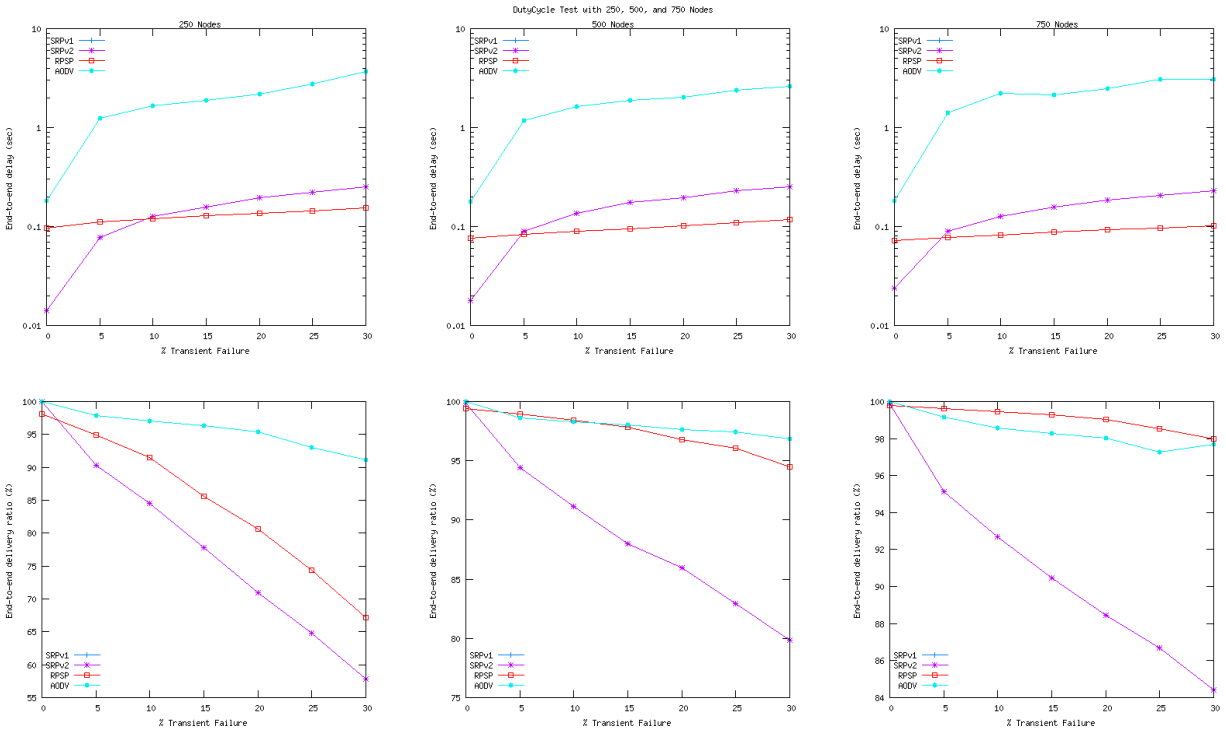

Fig. 5. Duty Cycle Test

model [19], 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.

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. Fig. 4 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 increases 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%.

RPSP makes an improvement over SRPv1 and SRPv2 in terms of end-to-end delay, as see in Fig. 4. It maintains a better end-to-end delay for all node densities.

The end-to-end delay significantly affects AODV, increasing significantly as 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.

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. Fig. 5 shows the results for the duty cycle test. For end-to-end delay, RPSP, as expected, has higher delay 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. RPSP additionally maintains roughly that same end-to-end delay no matter what the node density is 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 connectivity, as nodes fail. AODV performs poorly when the node density increases to 750 nodes, when there are a large number of transient failures.

6. Conclusions and Future Work

In this paper, 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. Fig. 3 above shows the best protocol in the SSRPF for each operating environment based on the simulation results show in Fig. 4 and Fig. 5. 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 delivery ratio metric. Overall, however, RPSP delivers the most reliable, fast communication using small 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. For this extension, the notion of the preferred path is valuable, as the nodes not on the preferred path do not need to listen to the packets, as long as the node on the preferred path is on. Thus, they can drop listening to a broadcast after hearing the header of the packet and listen to the entire broadcast only if the packet is rebroadcast and marked in the header as such. The second extension needs to address the challenge of efficient updates to hop distance to the destination. This 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. We plan also to introduce a time-to-live (TTL) on the hop distance in each node, after which the node would learn its distance from neighbor whose hop distance is still alive. The node's TTL in such a solution will be dependent on the speed with which the node moves.

7. References

[1] I.F. Akyildiz; W. Su; Y. Sankarasubramaniam; E. Cayirci. 2002. "A survey on sensor networks." In *IEEE Communication Magazine*, **40**(8):102-114.

[2] A. Woo; T. Tong; D. Culler. 2003. "Taming the underlying challenges of reliable multihop routing in sensor networks." In *Proc. ACM SenSys '03*, 14-27.

[3] J. Zhao; R. Govindan. 2003. "Understanding packet delivery performance in dense wireless sensor networks." In *Proc. ACM SenSys '03*, 1-13.

[4] G. Anastasi; A. Falchi; A. Passarella; M. Conti; E. Gregori. 2004. "Performance measurements of motes sensor networks." In *Proc. 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 174-181.

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

[6] J. Hill; R. Szewczyk; A. Woo; S. Hollar; D. Culler; K. Pister. 2000. "System Architecture Directions for Networked Sensors." In *Proc. 9th ACM Int. Conf. Architectural Support for Programming Languages and Operating Systems*, 93-104.

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

[8] W.R. Heinzelman; J. Kulik; H. Balakrishnan. 1999. "Adaptive protocols for information dissemination in wireless sensor networks." In *Proc. ACM MobiCom*, 174-185.

[9] C. Intanagonwiwat; R. Govindan; D. Estrin. 2000. "Directed diffusion: a scalable and robust communication paradigm for sensor networks." In *Proc. ACM MobiCom*, 56-67.

[10] G. Chen; J.W. Branch; B.K. Szymanski. 2006. "A self-selection technique for flooding and routing in wireless ad-hoc networks." In *Journal of Network and Systems Management*, **14**(3):359-380.

[11] G. Chen; J.W. Branch; B.K. Szymanski. 2005. "Self-selective routing for wireless ad hoc networks." In *Proc. of IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications, WiMob'05,* vol. 3, 57-65.

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

[13] R. Poor. "Gradient routing in ad hoc networks."

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

[15] A. Boukerche. 2009. *Algorithms and Protocols for Wireless Sensor Networks*, John Wiley and Sons Inc.

[16] B.K. Szymanski; C. Morrell; S.C. Geyik; T. Babbitt. 2008 "Biologically Inspired Self-Healing Routing with Preferred Path Selection." *Bio-Inspired Computing and Communication, LNCS, vol. 5151, Springer*, New York, NY, pp. 229-240.

[17] T. Babbitt; C. Morrell; B.K. Szymanski; J. Branch. 2008. "Self-Selecting Reliable Path for Wireless Sensor Network Routing." *Computer Communication Journal*, **31**(16):3799-3809.

[18] G. Chen; J.W. Branch; M. Pflug; L. Zhu; B.K. Szymanski. 2005. "SENSE: a wireless sensor network simulator," in *Advances in Pervasive Computing and Networking*, B. Szymanski and B. Yener, Ed. New York: Springer, 249-267.

[19] T. S. Rappaport. 2002. *Wireless Communications: Principles and Practice (2nd Edition)*, Prentice Hall.