

Self-Selecting Reliable Paths for Wireless Sensor Network Routing

Thomas A. Babbitt, Christopher Morrell, and Boleslaw K. Szymanski
Department of Computer Science, Rensselaer Polytechnic Institute
110 8th Street, Troy, NY
{babbitt, morrec, brancj, szymansk}@cs.rpi.edu

Joel W. Branch
IBM Research
12180 Hawthorne, NY
branchj@us.ibm.com

Abstract

Routing protocols for wireless sensor networks face two challenges. One is an efficient bandwidth usage which requires minimum delay between transfers of packets. Establishing permanent routes from the source to destination addresses this challenge since the received packet can be immediately transmitted to the next node. However, any disruption on the established path either causes packet loss, lowering the delivery rate, or invokes a costly process of creating an alternative path. The second challenge is the ability to tolerate permanent and transient failures of nodes and links, especially since such failures are frequent in sensor networks. Protocols that chose the forwarding node at each hop of a packet are resilient to such failures, but incur the delay caused by selection of the forwarding node at each hop of the multi-hop path.

This paper presents a novel wireless sensor routing protocol, Self-Selecting Reliable Paths (SRP) for Wireless Sensor Network (WSN) routing, that addresses both challenges at once. This protocol evolved from the Self-Selecting Routing (SSR) protocol which is essentially memory-less. In the first generation of SSR protocol each packet selects the forwarding node at each hop on its path from the source to destination. The protocol takes advantage of broadcast communication commonly used in WSNs as a communication primitive. It also uses a prioritized transmission back-off delay to uniquely identify the neighbor of the forwarder that will forward the packet. As a result, the protocol is resistant to node or link failures as long as an alternative path exists from the current forwarder to the destination. The second generation of SSR protocols, called Self-Healing Routing (SHR) added the route repair procedure, invoked when no neighbor of the forwarder closer to the destination is alive. In a series of transmissions, a packet trapped at the current forwarder by failures of its neighbors is capable of backing off towards the source to find an alternative route, if such exists, to the destination.

The main contribution of this paper is the third generation of SSR protocols, termed Self-Selecting Reliable Path, SRP. It preserves SHR's dynamic path selection in face of failure. Yet it also enables packets to follow established paths without selection delay if failures do not occur. The important change in the protocol is to make it memorize the successfully traversed path and attempt to reuse it for subsequent packets flowing to the same destination. The interesting behavior of SRP arising from this property is that if a path from the source to destination exists on which no transient failures occur, SRP would converge its routing to such a reliable path.

In the paper, we describe novel elements of the SRP protocol that resulted in the desired properties. Using simulation, we compare SRP protocol with the representatives of the two other approaches: AODV as the route-based protocol, and GRAB and SHR as the hop-selection protocols.

Keywords: routing protocol; wireless sensor networks; fault tolerant routing; reliable path selection; sensor network simulator.

1. Introduction

Wireless sensors networks (WSNs) consist of a number of sensor nodes connected through a wireless network, enabling a wealth of pervasive monitoring applications [1]. A significant number of them require battery-powered nodes to operate unattended and survive long periods of time (i.e., weeks to months) under less-than-ideal environmental conditions. Hence, research challenges involve building autonomy (self-management), fault-tolerance, and energy efficiency into all aspects of WSN operation. This especially applies to routing, since multi-hop communication is a primitive WSN operation that is extremely fault-prone as well as energy-intensive. For instance, commonly observed in WSNs are faulty (or, potentially subverted) nodes and transient and asymmetric links caused by wildly oscillating packet reception quality which cause severe packet loss and spontaneous network topology changes [2], [3]. Regarding energy usage, radio operation is typically the most costly hardware operation, as evidenced by a study in [4] and typical hardware specifications [5]. These challenges require a robust protocol for transmitting packets.

A traditional approach to multi-hop routing is to use routing tables that indicate which neighbor to forward the packet to in order to reach a destination; prominent examples include AODV [6], MintRoute [7], and Directed Diffusion [8]. This fundamental approach, which emulates traditional wired network communication, requires nodes

to constantly maintain individual neighbors' states (e.g., *active* or *sleeping*) to support routing decisions. Additionally, techniques for measuring wireless link conditions may be required as well [9]. Therefore, these types of routing protocols often require significant overhead to accommodate typical WSN operating conditions, 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.

In response to this challenge, we introduced the Self-Selective Routing (SSR) protocol [10,11,12] for wireless sensor networks in which the nodes within the range of the transmission of the current packet forwarder autonomously decide which one of them will forward the packet based on their hop distance to the destination. Since wireless sensor networks are often densely deployed, the major challenge in SSR design was to ensure uniqueness of the selection of the forwarding node to avoid duplicate paths that would increase energy consumed by communication. Despite the novel properties of the protocol, its design suffered from two problems. One was its inability to repair routes in which no alternative paths exist in the neighborhood of the fault. The second is shared with all routing algorithms that decide a path to take at each hop. Making such a decision introduces a delay at each hop resulting from an additional delay along the path and inefficient bandwidth use.

The first problem has been addressed in next generation of SSR protocols, called Self-Healing Routing, SHR [13,14], because of its ability to repair ("heal") broken routes. To achieve that, a node increases its hop distance to the destination if its repeated transmissions of a packet are not answered by any eligible neighbor. Overall, SHR dynamically and locally determines the shortest routes, even in the context of spontaneous topology changes. This also makes SHR a natural complement for energy-efficient topology control algorithms that control radio power states.

Routing protocols for wireless sensor networks face two challenges. One is an efficient bandwidth usage which requires minimum delay between transfers of packets. Establishing permanent routes from the source to destination addresses this challenge since the received packet can be immediately transmitted to the next node. Protocols of this kind include AODV [6], MintRoute [7], and Directed Diffusion [8]. However, any disruption on the established path either causes packet loss, lowering the delivery rate, or invokes a costly process of creating an alternative path. The second challenge is the ability to tolerate permanent and transient failures of nodes and links, especially since such failures are frequent in sensor networks. Protocols that chose the forwarding node at each hop of a packet are resilient to such failures. Hence, they are memory-less and do not retain any information about the paths taken by the same flow in the past. As a result, they incur the delay caused by selection of the forwarding node for each packet and at each hop of the multi-hop path. Protocols in this class include GRAd [15], GRAB [16], SSR [11] and SHR [14]. They avoid the use of geographical location information. GRAB also uses a more aggressive fault-tolerance technique by allowing redundant packets to follow multiple paths to a destination. SSR forgoes this approach and relies strictly on its prioritized transmission back-off delay technique to support (limited) fault-tolerance. SHR also avoids duplication of packets and extends SSR's fault tolerance by providing a route repair algorithm.

This paper presents a novel wireless sensor routing protocol, Self-Selecting Reliable Paths (SRP) for Wireless Sensor Network (WSN) routing, that addresses both above mentioned challenges at once. The main contribution of this paper is the description and analysis of the route repair introduced in SHR and definition and evaluation of performance of SRP. This new protocol preserves the SHR's dynamic path selection in face of failure. Yet it also enables packets to follow an established path without selection delay if failures do not occur. The important change in the protocol is to make it memorize the successfully traversed path and attempt to reuse it for subsequent packets flowing to the same destination. The interesting behavior of SRP arising from this property is that if there exists a reliable path from the source to destination on which no transient failures occur, SRP would converge its routing on such a path. More precisely, it will converge on the shortest reliable path.

The remainder of this paper is organized as follows. Section 2 describes our research background in Self-Selective Routing. The new contributions to the protocol design, mainly the improvements from SHR to SRP are presented in Section 3. Section 4 presents a comparison of SRP with other protocols using SENSE simulator [17], and a short subsection is dedicated to its current use. Using simulations, we compare SRP protocol with the representatives of the two other approaches: first with GRAB and then with AODV and SHR. The related work is discussed in Section 5. Finally, Section 6 provides conclusions and describes potential future research on Self-Selecting Routing.

2. Research Background

In all protocols of SSR family, each node knows its distance, in the number of hops, from a destination node. This distance is established via an initial route request and route reply stage. In this paper, 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:

$$d_{backoff} = \begin{cases} \lambda \cdot ((h - h_{expected} + 1) \cdot U(0,1)) & \text{if } h > h_{expected} \\ \frac{\lambda}{h_{expected} - h + 1} \cdot U(0,1) & \text{if } h \leq h_{expected} \end{cases}, \quad (1)$$

where 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 (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 for which it is waiting to transmit, it will cancel this transmission. Upon hearing the packet being transmitted, the sender will also send an *acknowledgement (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 (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 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 (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 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 (1), node B's delay will be $d_{B_backoff} = \lambda \cdot U(0,1)$ and node D's delay will be $d_{D_backoff} = 2\lambda \cdot U(0,1)$. The probability that node D will choose to forward the packet is then

$$p = \int_0^\lambda \frac{\lambda - x}{\lambda} \frac{dx}{2\lambda} = \frac{1}{4} \quad (2)$$

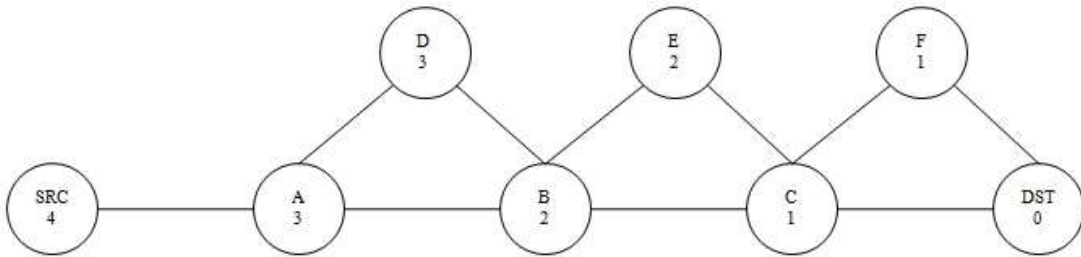


Figure 1. An illustration how a packet may travel a longer than necessary route under SSR

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 (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 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.

2. SHR

Here, we describe the route repair routine introduced in SHR protocol, which improves upon the aforementioned deficiencies of the original SSR protocol.

2.1.1 Route Repair

First, upon receiving a DATA packet, instead of using Equation (1), a node will ignore the packet if its hop distance is larger than the expected hop distance of the packet plus retransmission bit, and otherwise it will use the following equation to determine the delay before forwarding the packet:

$$d_{backoff} = \frac{\lambda}{h_{expected} - h + 1 + retransmission} U(0,1) \cdot \quad (3)$$

As the name indicates in Equation (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 (1), delays computed according to Equation (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) 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 further from the destination than itself. In this case, corrective action must be taken to reroute packets along the remaining shortest route.

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 nodes 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 effected flow.

An example of the route repair route is given in Figure 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 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 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 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 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 2(d)). In this scenario, the next packet from the source will find only alternative route via nodes E, F, G, and H, completing the route repair and sending this packet on the route to the destination¹. From this point on all packets will travel along the new path.

¹ In this implementation, three packets are lost before the route is repaired. We believe that this loss could be avoided if the original sender (node B in Figure 2(b)) is allowed to respond to rebroadcast if the new hop count is received (despite that it sent this packet to the sender, node C), enabling the original packet to complete the full route repair.

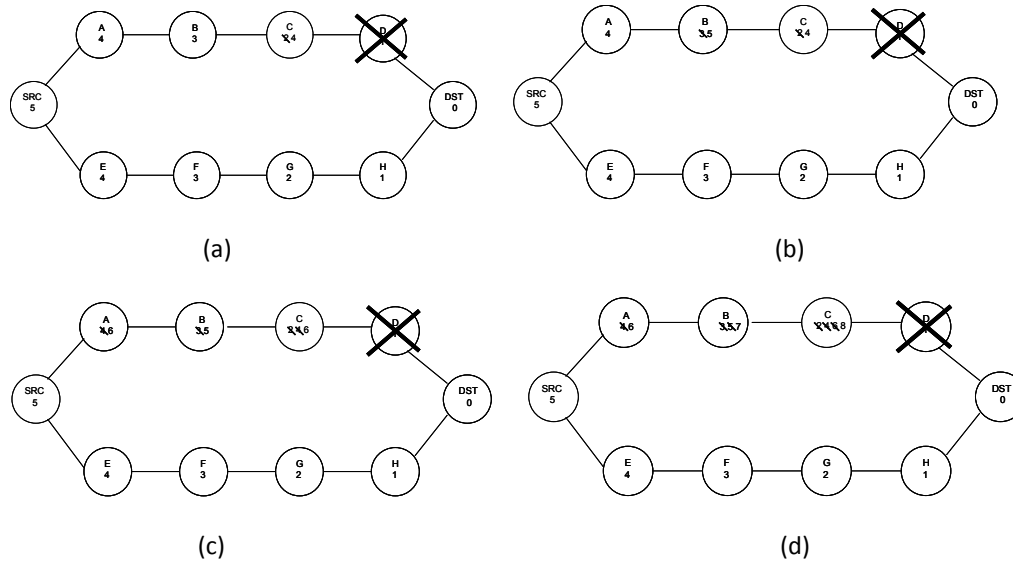


Figure 2. SHR route repair scenario

2.2 Route Repair Cost

Although the route repair was initially reported in [13], its costs or even convergence was not established. Here, we establish a bound on the cost of route repair in SHR and therefore also in SRP which inherited this algorithm. 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 will call such a step a recalibration of the hope 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 l , so at most $(n-1) * n/2$, hence $O(n^2)$ recalibration steps are needed.

To show that this limit is tight, let's consider a network consisting of two separate lines, each of $n/2-2$ nodes connected to the source and the destination (Figure 2 contains example of such network for $n=12$). Let's assume that one line is cut off from the destination at the last hop (as shown in Figure 2). It is easy to show by induction on the size of the network, n , that $n(n-2)/8$ recalibrations and $n(n-2)$ hops are required to recalibrate this network after such failure. Once the first packet recalibrates the network, all the subsequent ones would be able to follow the shortest surviving path.

To find a bound on the number of hops that such a recalibrating packet does, we notice that a recalibration must happen at least every n hops. Indeed, each hop without the recalibration decreases the distance of the packet to the destination by 1 , so after n hops the packet would arrive there. From this observation, it immediately follows that the number of hops made by the packet recalibrating the network after a failure is less than the cube of the number of nodes in the network.

3. SRP

GRAB, SSR and SHR are essentially memory-less. No matter how many packets traversed the route from the source to destination, the subsequent packet establish this route at each hop on the way anew. In SHR, this is motivated by the desire to maintain reliability of the routing, and therefore all neighboring nodes of the current

packet owner that are one hop closer to the destination are ready to forward the packet. Allowing all of them to forward the packet immediately, not only will create multiple paths, but also a collision as all nodes attempt to transmit causing large time delay and excessive energy use. We may say that all such neighboring nodes are fully symmetric to each other. To break this symmetry, SHR uses random delay. The extent of the random delay is defined by a coefficient λ . As shown in [18], the average delay on a single hop with n nodes competing for forwarding the packet is $\frac{\lambda}{n+1}$ while the probability of collision of two responses is $\frac{ns}{\lambda}$, where s is the minimum time needed for the node to discover that the medium is not free and avoid collision by suppressing its own transmission. Hence, the average number of packets that needs to be transmitted to get collision-free selection is $\frac{\lambda}{\lambda - ns}$ and therefore we must set $\lambda \gg ns$. The typical value of s is today at about 0.1 ms and n , the number of neighbors of the node is at most ten, so reasonable values of λ are in tens of ms. Clearly there is a tradeoff between the delay and probability of collision that impacts the quality of routing in SHR and the delay of tens of ms per hop is inherent in this protocol.

To address this concern, we have changed the protocol by breaking the symmetry between responding nodes based on previous packet forwarding. One, and only one node forwarded the previous packet and that node is allowed to forward the current packet immediately. Although simple, this rule, deeply transformed the performance of the protocol. No longer is the protocol memory-less. Now, the successful traversal from the source to destination creates a semi-permanent path. As long as all the nodes and all the links on this path are reliable, the packets will follow this path, that we call a reliable path, with no delay (a miniscule delay is added to check if the medium is free) at any hop. However, if any of the nodes or links fails, all the neighbors of the node preceding the failed link or node stand ready to take over forwarding the packet. Hence, this approach combines the best parts of two routing philosophies. The first is a rigid approach using network routing tables that must be updated as changes occur. Yet, thanks to the static routes, in a stable network, this approach enables the receiving node to transmit immediately knowing that it is the only one allowed to do so. The second approach relies on dynamic routing decisions which works well in unstable networks, but requires back off delays to ensure that only one node closer to the destination forwards the packet. The selection of such unique node delays packet forwarding. Hence, as more traffic or source nodes are introduced into the network, the delay imposed by waiting for the selection of the forwarding node can cause excessive collisions, slow flow of packets, packet loss, and excessive energy use for packet retransmissions.

The interesting behavior of this protocol arising from the way it selects its routes is that if there exists a path from the source to destination on which no transient failures occur, the protocol will converge its routing to such reliable path. Even more, it will converge to the shortest reliable path. Here is the proof.

Let's 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

$$c_{ave} = p_s \sum_{i=1}^{\infty} i \cdot (1 - p_s)^{i-1} = \sum_{i=1}^{\infty} (1 - p_s)^{i-1} = \frac{1}{p} = 1 + \frac{m_t}{m_s}. \quad (4)$$

If there is a stable path at all, through route repair, it will be selected after the finite number of packets flow through, because even if a path with transient links were selected initially, there is a nonzero probability that all the possible forwarder fail to respond twice in a row, initiating a route repair, which will eventually end up forcing the flow through the shortest stable existing path.

Accordingly, we called this protocol Self-Selecting Reliable Path routing, or SRP in short.

3.1 SRP Finite State Automata

The actions of a node upon receiving a DATA packet are difficult to verify and implement correctly. Hence, we are using a Finite State Automata (FSA) to encapsulate these actions and the protocol is hand-coded

based on this FSA. Figure 3 depicts the states that a node goes through as it receives and processes a packet. For each flow in which a node participates, there is a different automaton possibly in a different state defining the actions of the node. Using an FSA aided design, debugging and documenting the actions of the protocol. The details of the state transitions of this FSA are available at <http://www.ita.cs.rpi.edu/sense/>.

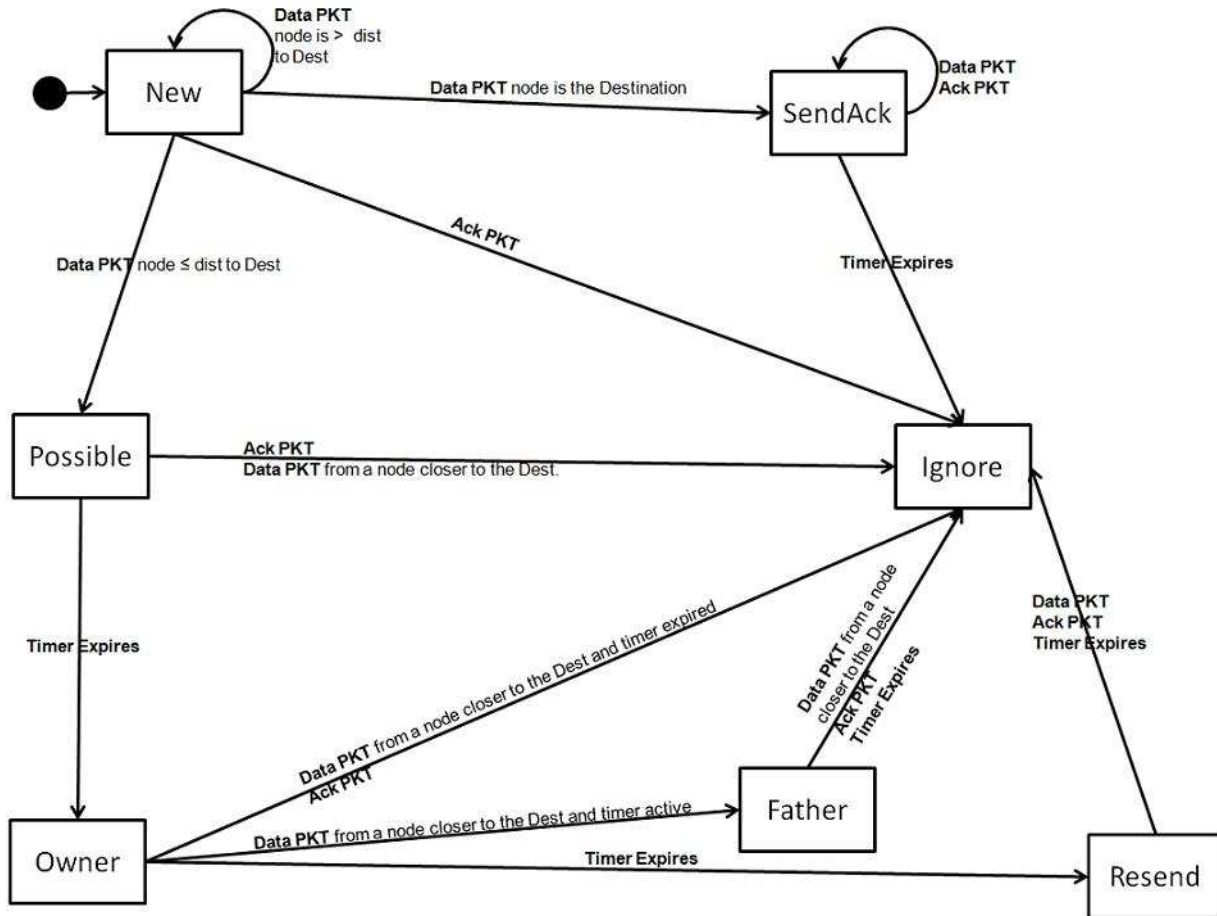


Figure 3. SRP state diagram

4. Performance Evaluation by Simulation

4.1. SENSE Sensor Network Simulator

To evaluate and verify the presented protocols, we used the SENSE simulator [17] that is an easy to use, efficient, and powerful sensor network simulator freely available at <http://www.ita.cs.rpi.edu/sense/>. Three critical factors went into the design of SENSE: extensibility, reusability, and scalability [19].

Since its first publication and use in simulating SSR, it has gained some following at many places in the world, as evidenced by simulating results of WSNs reported, according to Google in 36 papers. For example, in [20] the authors simulate *C2E2S* (Cluster and Chain based Energy*Delay Efficient Routing Scheme) for wireless sensor networks. In [21], the authors used SENSE to simulate Coordinate-based Data Dissemination protocol (CODE) and Sink Cluster-based Data Dissemination protocol (SIDE). Hence, we feel that SENSE is the reliable tool for comparison of the performance of our protocols with others.

4.2. Comparison with GRAB

We conducted a number of simulations to compare SRP to the published results of GRAB in [12]. We conducted these simulations to show that SRP could compete against another protocol developed specifically for WSN that uses 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 [16]. Using SENSE, we conducted a series of

simulations to mimic the ones published in [16] 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 $150 \times 150\text{m}^2$ 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 the total of 100 packets. The nodes were abstraction of the Berkeley nodes [22], which consist of a 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 10 ms 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 free space signal modal gave similar results. The reported results were averaged over 10 simulation runs.

To match settings under which those results were obtained, we simulated performance of SRP under both the density test and the permanent failure test. We created a 15 unit by 15 unit terrain populated with 1200 nodes; each node is stationary, and has a single unit nominal transmission range. We sent packet every 10 seconds and ran the simulation 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 [16] 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 [5,14] 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 [16]). Selecting transient link in our simulation, we have not considered physical distance from the sender. In real deployment, most transient links are at the far edge 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 dropping 90% of packets they overhear, we effectively lose 15% of the packets at the node level.

4.2.1 Density Test

In the density simulation we set the permanent failure and link failure rate to 15%. We ran ten simulations for each density level from 600 to 1800 nodes, similarly to the simulations reported in [16]. Nodes that fail as part of the 15% permanent failure are randomly chosen and uniformly distributed throughout the simulation. The results for the density test show that SRP is considerably more effective than GRAB in sparse network topologies, as depicted in Figure 4.

For 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.

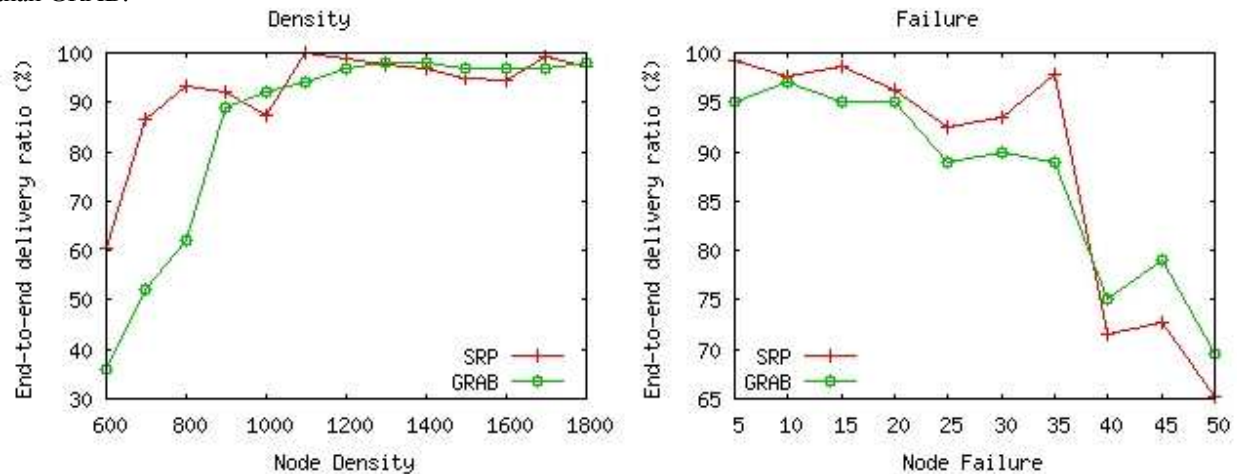


Figure 4. Comparison between SRP and GRAB under density and permanent failure test with total of 100 packets sent

4.2.2 Failure Test

In the permanent failure simulations we set the transient link failure rate of 15%. We ran ten simulations for each permanent failure rate from 5% to 50% in increments of 5% to get the results comparable to those reported in [16] with the configurations set as 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 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 delivery rate of 95% exceeded 89% range of GRAB. Both protocols maintain over 95% delivery rate when permanent failures are less than 20% frequent.

As has been discussed throughout the paper, SRP attempts to take advantage of both: (i) dynamic route selection similar to the way GRAB and SSR select paths from source to destination, and (ii) 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 cause considerable turbulence with a test length of only 100 packets. However, when a semi-stable route can be found, even for a period of time, the reliable path (static route) is quickly established and taken advantage of. This route will quickly send the packets. 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. As will be described below in Section 4.3, when simulations are run longer than for 100 packets the delivery rate of SRP, even with a 50% permanent failure rate, is considerably higher.

4.3. Volume Test

While conducting the tests to compare SRP to GRAB we conducted some simulations under the same constraints used in the failure and density tests described in Section 4.2 above, but increased the run time for the simulations. Additionally, we simulated sending packets at a faster rate than 10 seconds in an effort to see how SRP withstands higher volumes of traffic. We conducted these tests because 100 packets is a small sample size and one bad test out of 10 can significantly alter results. We noticed that in both instances of the longer runs, the delivery rate increased indicating that SRP successfully handles the increased traffic load. So we conducted three complete sets of failure and density tests exactly as described in section 4.2 with the exception that we sent a packet every 10 seconds, 1 second, and 0.3 seconds. We ran each simulation for 10,000 seconds resetting the statistics at 1,000 seconds to remove the network initialization period.

We cannot compare this results to those published in [16] because of lack of access to more thorough tests of GRAB. So the charts below are only a comparison of SRP to itself with higher volumes of traffic and for longer time periods. The point is to illustrate that SRP indeed combines reliability of the dynamic route selection with the speed of static path routing. This combination of benefits enables SRP to perform reliably over a long time periods and with high traffic rates.

4.3.1 Density Test

Figure 5 shows the results for the three density simulations. Each was conducted with the same configurations as the comparison with GRAB, except that simulations sent packets at 10 second, 1 second, and 0.3 second intervals. They each had 15% link and permanent failure rates. Each test was 10,000 seconds long with the statistics cleared at 1,000 seconds. The average of 10 tests was used at each permanent density level.

The results in Figure 5 for the simulation with a 10 second interval compared with the results in Figure 4 show that extending the simulation from 100 packets to 10,000 seconds improved the delivery rate for a density of 600 nodes from 60.6% to 66.8%. The reason is simple. During each simulation, 15% of the nodes fail. These nodes are chosen randomly and they fail uniformly over the time of simulation. In total, the same number of nodes will fail in a 1,000 second and a 10,000 second simulation, if all other constraints are the same. Each failure might cause a route recalculation in SRP. On average, the same numbers of recalculations are required in both simulations and so they have a constant cost. The big difference is the total number of packets sent. The number of hops backward that the route has to repair will dictate the number of packets lost. However, in a simulation with 600 nodes, each node has very few neighbors so it is likely that a route has to repair several hops. On average a loss of approximately 2-5 packets occur. In addition, there is likely to be some routes that rely on transient links which cause additional route corrections not normally needed in a denser network. If there is an alternate route, SRP will find it and will send packets along it.

For all density levels, the delivery rate of packets increases as the time interval between packets decreases. This is because the number of packets transferred in the simulation increases too, and the network quickly converges to the stable routes. This convergence to the stable routes and route repairs in response to failures impose the

constant cost of lost packets that is amortized over the increasing number of packets transferred. This, however, means also that SRP efficiently transfer high volume traffic along stable paths, as it was intended to do so. The key observation is that even in very sparse networks (with 600 nodes), when sending packets at 0.3 seconds, the delivery rate is 92.9%. This illustrates how quickly and efficiently SRP adjusts to route failures and finds alternate routes with minimal packet loss, even in sparse networks.

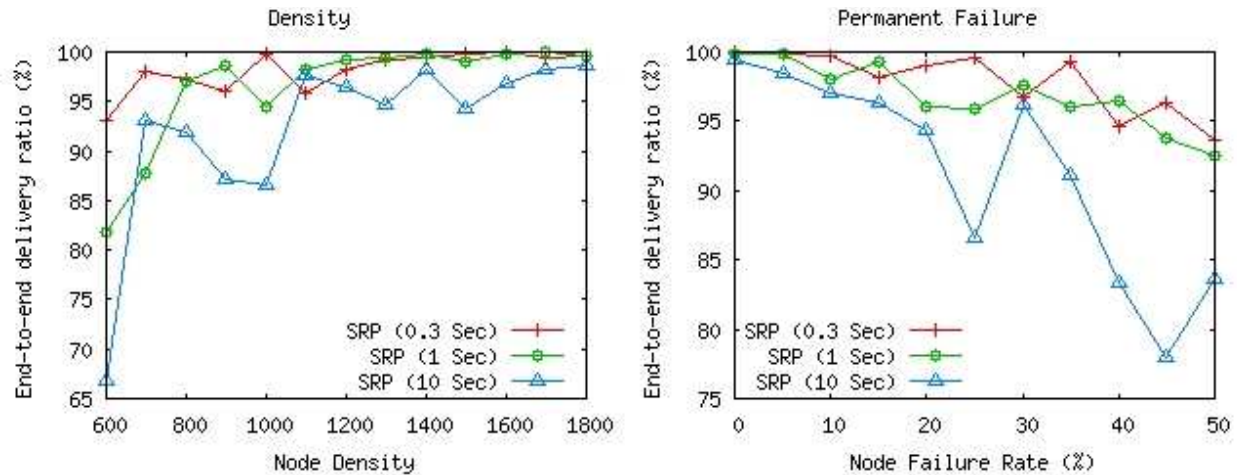


Figure 5. SRP performance on density and permanent failure tests with different transmission rates

4.3.2 Failure Test

Figure 5 shows the results for the three permanent failure simulations. Each simulation used the same configuration as the tests to compare SRP to GRAB with the only modification of sending packets at 10 second, 1 second, and 0.3 second intervals. Each simulation had constant 15% link failure rate and permanent failure rates changing from 5% to 50%. Each test was 10,000 seconds long with the statistic cleared at 1,000 seconds. The average of 10 tests was used at each permanent failure rate.

Comparing the results in Figure 5 for the failure simulation at 10 second interval with the failure test above in Figure 4 that sent only 100 packets, it is clear that running the simulation longer significantly improves the results. At 50% failure in Figure 4, the delivery rate was 65.2% whereas running the same simulation for 10,000 seconds, as seen in Figure 5, yielded an 83.7% delivery rate. When a path becomes blocked in SRP, the route repair algorithm is executed. Depending on how many hops backward needs to be adjusted, route repair can lose from one to many packets fixing the route. On average, especially in networks that have significant permanent failures, 2-5 packets are lost fixing the route. Hence, the packet losses are constant in both simulations because both have the same number of permanent failure. However, those losses are spread over the large number of packets being sent in the longer simulation, improving the delivery rate.

The delivery rate increases as the time interval between packets decreases. This can again be explained as in the previous subsection, so we merely observe that even at a 50% node failure rate, when packets are sent at 1 second or 0.3 seconds the delivery rate is over 90%. This again illustrates speed with which SRP adjusts to route failures.

4.4. SRP Compared to Static and Dynamic Path Protocols

To enable full comparison of SRP with the protocols representing two competing approaches, static path routing and dynamic path selection, we selected AODV as the representative of the former and SHR as the example of the later. All three compared protocols were implemented using the SENSE simulator. AODV is a well-accepted and well-documented wireless routing protocol whose reliance on established paths is very similar to other traditional protocols. Hence, it serves as a good baseline comparison against a protocol that maintains routing tables.

We tested three different scenarios. The first one involved a single sink (base station) collecting data from many sources, a typical sensor network setting. The second scenario considers transient failures which can be prevalent especially during energy shut down cycles or in presence of transient links. The third considers permanent failures which are likely to occur as nodes batteries die or the nodes are destroyed by their environment.

For all simulations, the topography consists of an 8 unit by 8 unit terrain populated with 500 nodes placed randomly. Each node is stationary, and has a single unit nominal transmission range. The wireless medium is simulated with the free space propagation model [23], and the radio modeled operation at 914 MHz with 1 Mb/s of

bandwidth abstracting the parameters of Crossbow nodes. Packet sizes are uniformly distributed around the mean of 1,000 bytes and are sent at uniformly distributed intervals with a mean of 40 sec. MAC broadcast was used in which a node senses the carrier and sends the packet only if no other transmissions are detected. The average hop distance between the source and destination is 7.8. Each simulation was executed ten times, each time with a different random number seed, and each for a simulation time of 3,000 seconds. The same 10 seeds were used for all simulation sets. λ was set to 100ms.

The failure sensitivity of SRP's route repair routine can be tuned by adjusting the number of packets required to invoke route repair. By increasing this value, SRP can be successfully employed in a network with a high rate of transient failures while still maintaining strong performance in a network with a high rate of permanent failures. In our tests, two packets were required to invoke route repair.

4.4.1 Single Sink Test

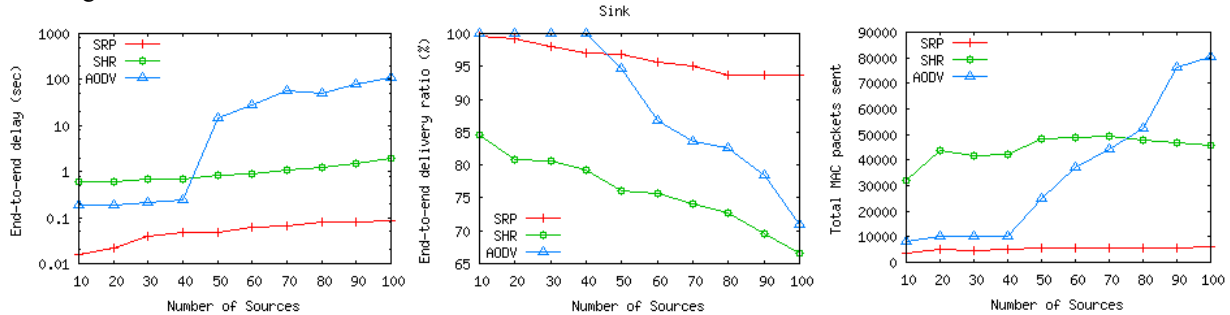


Figure 6. Transmission delay, delivery rate, and total MAC packets sent in the case of a single sink network for three compared protocols: AODV, SHR and SRP

In a wireless sensor network, a single sink network is common setting. For example, any network that contains a single base station is such. The sink topology may result in a heavy congestion of traffic near the sink node. Such congestion may cause massive amounts of collisions, and essentially stop a network from functioning at all. In sink network simulations, we varied the number of sources transmitting to a single sink from 10 to 100 to check scalability of each protocol tested. As is apparent in Figure 6, a single sink network is where SRP shows its true worth, and where AODV breaks under its limitations of having to update its routing table. SHR breaks having to dynamically create a path at each hop, resulting in transmissions delayed by competition for path selection.

The protocols' end-to-end delays were so drastically different, that a logarithmic scale was necessary. As the density of sources increases from 70 to 100, which is 14% to 20% of the nodes in the network, AODV required approximately 100 seconds and SHR requires approximately 1 second to transfer a packet from the source to the destination. Although SRP does increase its delay slightly, it still manages to keep that delay to under 0.1 seconds, even with 100 nodes transmitting. Clearly, the reliable path selection allows packets to move across the network quickly enough that a packet reaches the destination before the packets from other flows are transmitted. SRP is also superior in terms of delivery rate. As sources increase to 100, SRP's delivery rate only decreases slightly, while AODV's and SHR's drops to nearly 60%. The reason is that AODV and SHR succumb to the congestion around the sink node, while SRP's reliable path traversal is fast enough to avoid any congestion. SRP sends considerably fewer MAC packets than both SHR and AODV. This results in a reduction of energy use.

4.4.2. Transient Failure Model

In the transient failure model, each node was assigned a mean active time and a mean sleep time. The sum of these two times was fixed at 200 seconds. The time spent in each mode was distributed exponentially about the mean value.

There are several possible causes for transient node failures such as error-prone links, power management induced duty cycles, or excessive packet collisions. Of these, the duty cycle induced failures are the least disruptive since they may be coordinated with the networking protocol. The presented simulation results are based on a random transient failure model, so they exaggerate the effect of duty cycles on the protocols.

As seen in Figure 7, AODV has the worst transmission delay that increases significantly with the transient failure rate. This is caused by constant updates of the network routing tables. SRP and SHR, that do not maintain such tables, have lower delays than AODV for all cases in which transient failures are present. SRP has also faster transmission times than SHR thanks to its reliable path routing.

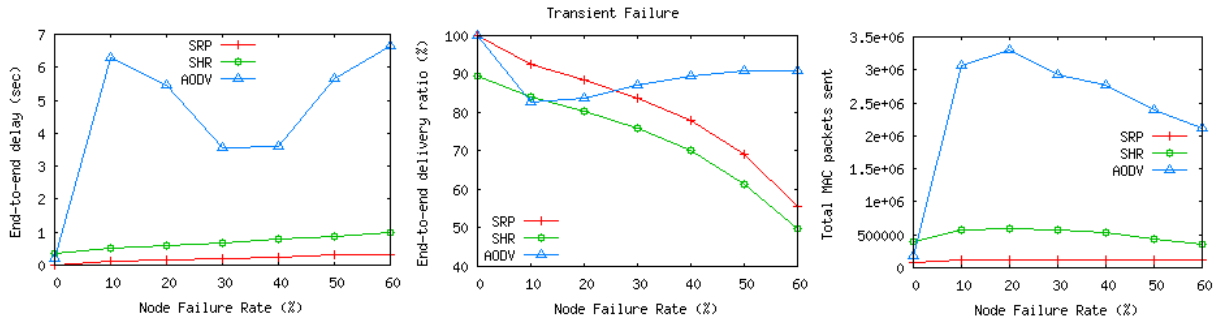


Figure 7. Transmission delay, delivery rate, and total MAC packets sent in the case of transient failures for three compared protocols: AODV, SHR and SRP

AODV’s delivery rate is the best, dropping from 100% in a reliable case to 95% for 30% transient failure rate, while SRP delivery rate drops from 100% to 76% over the same region and SHR’s is even slightly lower, dropping from 92% to 75%. However, AODV requires a much greater number of MAC packet transmissions than SHR and SRP. This is because the AODV’s route repair algorithm initiates a new route request phase for each failure, causing a flood of packets from the point at which the route is severed in search of a new path. AODV uses over 30 times more packets than SRP does.

4.4.2. Permanent Failure Model

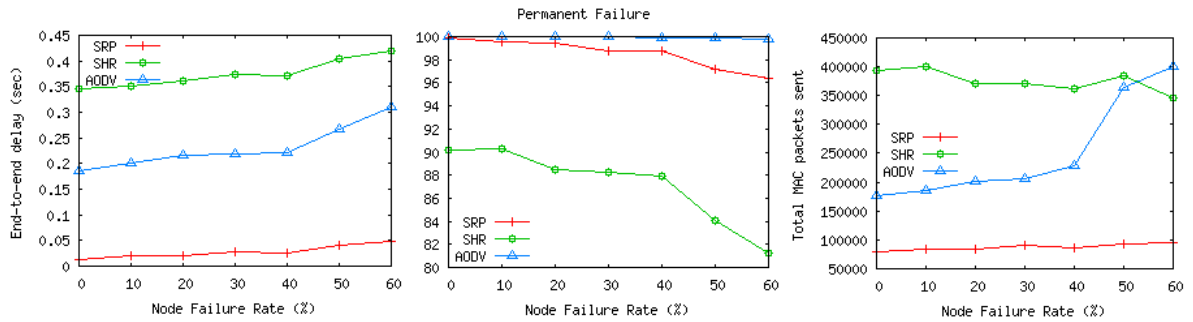


Figure 8. Transmission delay, delivery rate, and total MAC packets sent in the case of permanent failures for three compared protocols: AODV, SHR and SRP

In the permanent failure model, each node had a random chance of failing. Nodes that failed had their failure time uniformly distributed over the remaining simulation time. In this scenario, trends observed for transient failures continue but are less pronounced. As seen in Figure 8, when the number of node failures increases, the transmission delay and the delivery rate generally increase and decrease, respectively.

SRP achieves the lowest and most stable transmission delay of all three protocols. Even at 30% failure rate, its delay is only slightly increased compared with its delay in the reliable network, and is nearly 10 times better than that of AODV. Although SRP delivery rate is not 100% as is AODV’s, it still shows a 10% improvement over SHR, and stays at or above 98%. This improvement is based on the reliable path selection. In SHR dynamic path selection introduces delays into data transmission that cause unnecessary packet loss. SRP excels also in the total MAC packet sent. As failures increase, the number of packets required for AODV to maintain 100% delivery rate begins to quickly increase and finally, at 30% permanent failure rate matches the constant but high level of packets required by SHR. In contrast, SRP maintains leveled performance at approximately 30-40% of the total MAC packets used in AODV.

5. Related Works

Other opportunistic protocols rely on geographic location information to support routing decisions. For instance, BLR [24] uses location coordinates to allow only receivers in an “eligibility region” (defined as a region in which all nodes are closer to the destination than the sender and all can overhear each others’ transmission) to contend to forward packets. A prioritized back-off delay scheme, similar to SSR, ensures that the closest node forwards the packet and suppresses redundant transmissions. However, upon learning the closest receiver, the sender

will then forward following packets only to that receiver for a set number of transmissions. This latter technique may only be effective with ideal link qualities. GeRaF [25] also employs a similar eligibility region with a prioritized back-off delay technique. However, GeRaF uses a dual-radio approach with busy-tone signaling to make sure channels are clear before sending data to reduce the probability of collisions. GeRaF also uses a request-to-send/clear-to-send (RTS/CTS) packet forwarding technique which imposes higher packet forwarding overhead than SSR's approach. IGF [26] is similar to the above protocols, using an eligibility region defined as a 60° fan-shaped region extending from a sender directly towards the destination. If the sender does not hear a response from any nodes, it will shift the eligibility region and try to find other receivers. Other similar location-based protocols include PSGR [27] and SIF [28].

6. Conclusion and Future Works

In this paper, we have presented SRP, which leverages the fault tolerance of dynamic path selection in WSNs with the speed of a static path routing. Under SRP, the intermediate nodes on route to the destination use broadcast communication and prioritized transmission back-off delay to make packet forwarding decisions based only on their hop distance from the destination and their previous forwarding decisions. The sender of the packet listens to the responses, and if there is no response to the original and repeated transmissions (a clear sign of faults in its neighborhood), it increases its hop distance to the destination to facilitate seamless route repair. Additionally, SRP improves the end-to-end delay of the protocol by allowing the node that has won the self-selection to cut its back-off delay to zero for subsequent packets of the same flow. This node then is assured of winning the subsequent self-selections until failure of the node or its link removes it from the competition (in which case its back-off delay is restored to the original value). This design creates a protocol that is fast when there are no failures and yet robust when failures do occur.

We have identified several future directions for enhancing SRP. We intend to modify the route repair routing in order to avoid losing packets while correcting faulty paths. Currently, SRP uses random selection of a unique neighbor which will forward a packet if the reliable path is not yet established. This helps balance network energy usage over long time and over many flows, but the behavior is not explicit. The limits of SRP's current random selection behavior become apparent when either there are nodes that consume more energy than others (e.g., due to either long flow transmitted over the stable reliable path or excessive retransmissions, or even or due to heterogeneous device characteristics) or fresh nodes are dropped into the network. In either case, nodes with more energy should bear more of the routing burden in order for the network to consume energy in a balanced manner. Therefore, we intend to extend SRP's self-selection policy to give higher transmission priority to nodes that in addition to being closer to a destination than the sender, also have more energy than other neighbors.

We also mentioned that SRP can already accommodate topology changes, which could possibly be caused by energy-efficient topology control algorithms (e.g., GAF [29] and ESCORT [30]). However, the challenge with explicitly incorporating a topology control algorithm into SRP is making sure that the algorithm is not so aggressive that it negatively affects SRP's fault-tolerant behavior by severely reducing the number of available receivers. Hence, we are researching how to provide a dynamic topology-control policy for SRP.

Acknowledgments

The authors thank Gilbert G. Chen of RPI for his insight on this paper as well as his contributions to earlier research on this topic and Geoff Merrett of the University of Southampton in the United Kingdom for his suggestion of the need for route repair routines.

This work was sponsored by US Army Research laboratory and the UK Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

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] 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] W.R. Heinzelman; J. Kulik; H. Balakrishnan. 1999. "Adaptive protocols for information dissemination in wireless sensor networks." In *Proc. ACM MobiCom*, 174-185.
- [8] C. Intanagonwiwat; R. Govindan; D. Estrin. 2000. "Directed diffusion: a scalable and robust communication paradigm for sensor networks." In *Proc. ACM MobiCom*, 56-67.
- [9] Y. Xu; W.-C. Lee. 2006. "Exploring spatial correlation for link quality estimation in wireless sensor networks." In *Proc. IEEE PERCOM'06*, 200-211.
- [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] G. Chen; J. Branch; B.K. Szymanski. 2005. "Local Leader Election, Signal Strength Aware Flooding, and Routeless Routing." In *Proc. 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc Networks and Sensor Networks, WMAN05*.
- [13] J.W. Branch; M. Lisee; B.K. Szymanski. 2007. "SHR: Self-Healing Routing for wireless ad hoc sensor networks." In *Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS'07*, 5-14.
- [14] 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*.
- [15] R. Poor. "Gradient routing in ad hoc networks."
- [16] 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).
- [17] 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.
- [18] B.K. Szymanski; G. Chen. 2008. "Computing with Time: From Neural Networks to Sensor Networks." In *Computer Journal*, **51**, in print.
- [19] G. Chen, B.K. Szymanski. 2001. "Component-Oriented Simulation Architecture: Towards Interoperability and Interchangeability." In *Proc. Winter Simulation Conference'01*, 495-501.
- [20] TT Huynh; CS Hong. 2006 "An Energy* Delay Efficient Multi-Hop Routing Scheme for Wireless Sensor Networks." In *IEICE Trans. Inf. & Syst.*, E89-D(5):1654-1661.
- [21] H Le Xuan; Y Lee; S Lee. 2005 "Two Energy-Efficient Routing Algorithms for Wireless Sensor Networks." In *Proc. 4th International Conference on Networking, ICN*, 698-705.
- [22] 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.
- [23] T. S. Rappaport. 1996. *Wireless Communications: Principles and Practice*, Prentice Hall.
- [24] M. Heissenbttel; T. Braun; T. Bernoulli; M. Waelchli. 2004. "BLR: beaconless routing algorithm for mobile ad hoc networks." In *Elsevier's Computer Communications Journal*, **27**(11).
- [25] M. Zori; R.R. Rao. 2003. "Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: multihop performance." In *IEEE Trans. Mobile Computing*, **2**(4):337-348.
- [26] B. M. Blum; T. He; S. Son; J.A. Stankovic. 2003. "IGF: a robust state-free communication protocol for sensor networks." Technical Report CS-2003-11, University of Virginia.
- [27] Y. Xu; W.-C. Lee; J. Xu; G. Mitchell. 2005. "PSGR: priority-based stateless geo-routing in wireless sensor networks." In *Proc. IEEE Conf. on Mobile Ad-hoc and Sensor Systems*.
- [28] D. Chen; J. Deng; P.K. Varshney. 2005. "A state-free data delivery protocol for multihop wireless sensor networks." In *Proc. IEEE Wireless Communications and Networking Conf.*
- [29] Y. Xu; J. Heidemann; D. Estrin. 2001. "Geography-informed energy conservation for ad hoc routing." In *Proc. ACM MobiCom*, 70-84.

[30] J.W. Branch; G. Chen; B.K. Szymanski. 2005. "ESCORT: Energy-efficient Sensor network COMMunal Routing Topology using signal quality metrics." In *Proc. 4th Int. Conf. on Networking*, Springer-Verlag LNCS, vol. 3420, 438-448.