Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS'07
San Diego, CA, July 16-18, 2007, pp. 5-14.

SHR: Self-Healing Routing for wireless ad hoc sensor networks

Joel W. Branch, Mark Lisee, and Boleslaw K. Szymanski
Rensselaer Polytechnic Institute
Department of Computer Science
110 8th Street
Troy, NY 12180
{brancj, liseem, szymansk}@cs.rpi.edu

Keywords: opportunistic routing, wireless sensor networks

Abstract

This paper presents a novel protocol, Self-Healing Routing (SHR), for opportunistic multi-hop wireless communication. The protocol uses broadcast communication and a prioritized slotted transmission back-off delay scheme to empower a receiving node to use its hop distance from the destination to autonomously decide whether to forward a packet. This enables dynamic traversal of the shortest available routes without requiring nodes to explicitly decide to which neighbors to forward packets. When severed routes are encountered, the protocol dynamically and locally re-routes packets so they traverse the surviving shortest route. The result, as shown by simulation data reported here, is an efficient fault-tolerant protocol that performs well even when the network topology changes spontaneously.

1. Introduction

Wireless sensors networks (WSNs) are composed of a large number of wirelessly networked nodes with sensing and processing capabilities and have enabled 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].

A traditional approach to multi-hop routing is to use routing tables that indicate the neighbor to which a packet should be forwarded to reach a destination; prominent examples include AODV [6], MintRoute [7], and Directed Diffusion [8]. This fundamental approach, which emulates traditional wired network communication, naturally requires nodes to constantly maintain, or *assume*, 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 and is therefore the subject of this paper.

This paper presents a new WSN routing protocol called Self-Healing Routing (SHR), which represents our continuing research in fault-tolerant WSN routing. In SHR, instead of deciding where to forward a packet, a sending node freely broadcasts it to all nodes within its transmission range. Receivers then autonomously decide whether to forward the packet using only knowledge of their hop distances from the destination and a prioritized transmission back-off delay scheme that ensures that the packet is forwarded by only one of the contending receivers. SHR achieves this using a time-slotting approach which (i) ensures that the currently shortest available routes are *always* followed, and (ii) reduces the probability of packet collisions. Additionally, SHR enables efficient and local repair of severed routes. 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.

The remainder of this paper is organized as follows. Section 2 describes our research background. Section 3 describes the new contribution of this paper, the SHR algorithm. Section 4 describes the complete SHR routing protocol. Section 5 presents a theoretical analysis of

SHR. Section 6 evaluates SHR's performance via simulations. Section 7 describes related works and Section 8 concludes the paper.

2. Research Background

SHR is an extension of Self-Selective Routing (SSR) [10], [11] in which 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 stages described in Sections 4.1 and 4.2, respectively). For the packet forwarding process, instead of sending packets to designated neighbors, a node *broadcasts* a packet with the expected length of the remaining path to the destination to *all* of its neighbors. They then use the *self-selection algorithm* to decide autonomously which node will forward the packet further using a prioritized transmission back-off delay scheme. In it, after a node receives a packet, it schedules further packet transmission after a random delay whose average is proportional to its distance from the destination. The transmission back-off delay is specifically determined by the following equation*:

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

where h is the node's hop distance from the destination, $h_{expected}$ is the sender's hop distance minus 1, $U(0,1)$ is a random number generator producing real values uniformly distributed between 0 and 1 (helping to randomize delays and 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 rebroadcasting a packet first. If a node overhears another node re-broadcasting the same packet for which it itself is waiting to broadcast, it will cancel its transmission of the packet. Upon hearing the packet being rebroadcast, the sender will also broadcast an *explicit acknowledgement (ACK)* packet signaling all nodes to cancel their transmissions, just in case the self-selected node's transmission is out of listening range of other 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 freely broadcast over all links and have a high probability of reaching the best available neighbor in one transmission. However, SSR exhibits several limitations.

First, as it is clear from (1), SSR calculates a node's transmission back-off delay in continuous (non-slotted) time, which, as will be shown later, has higher probability of packet collisions than a slotted time solution.

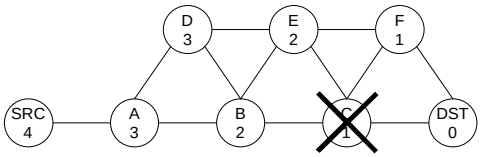

Figure 1. An example network illustrating how a packet may travel a longer than necessary route in the case of a topology change.

Second, with non-zero probability, delays based on (1) result in packets unnecessarily traveling longer routes while shorter routes are available in a realistic case where there may be node or link failures (being either transient or permanent), nodes that operate on duty cycles to save energy, or mobile nodes. For this paper, we assume static (non-mobile) nodes. Hence, each node may have neighbors that are only one hop closer, the same distance, or one hop farther from the destination than itself. With node or link failures there is no guarantee that any of the neighbors will respond. As a result, if, for a given broadcast, a node has no response from any neighbor that is closer to the destination than itself, delays generated according to (1) may result in a neighbor that is farther from the destination than the sender forwarding the sender's packet, sending a packet via a longer route. For example, consider the network shown in Figure 1, where nodes are represented by circles and their hop distances from the destination (labeled DST in the figure) are indicated by the numbers in the circles. Suppose that node B has forwarded a packet from node A with an expected hop distance of 1, but node C has not received it because of a faulty link, deactivated radio, or hardware failure. This leaves nodes D and E to *compete* for forwarding the packet (node A will not try to forward the packet since it just sent it). From (1), node D's delay will be $d_{D\_backoff} = \lambda \cdot (2 \cdot U(0,1) + 1)$, and node E's delay will be $d_{E\_backoff} = \lambda \cdot (U(0,1) + 1)$. The probability that node D will choose to forward the packet is then

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

Therefore, node B's packet has a one in four chance of following a route of length 7 instead of 6. 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, (1) can be improved to reduce such probability p and therefore enable better performance.

Third, SSR does not support any route repair routine for propagating packets around severed routes, which occur when a node has no available neighbor with a lower hop distance 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

* Equation (1) includes a correction to the equation for transmission back-off delay originally published in [10] and [11].

found in a way similar to the scenario in Figure 1. Relying on such backward travel is inefficient in two ways. One, since SSR enforces a *time-to-live* value for every packet, a packet's ability to find new routes as it travels farther back to its source is limited since the number of traversed hops increases. Two, SSR will not adapt its behavior in such a way as to *prevent* further packets from traveling down the severed route to the cutoff point.

3. Self-Healing Routing

Here, we describe our new contribution, the SHR algorithm, which improves upon the afore-mentioned deficiencies of the original SSR algorithm.

3.1. Prioritized Time-slotted Transmission Back-off Delay

The first improvement is the replacement of the original prioritized back-off delay scheme with a new time-slotted back-off delay scheme. Within this scope, two specific improvements have been implemented.

First, upon receiving a packet, instead of using (1), a node will use the following equation to determine the delay before forwarding the packet:

$$d_{backoff} = \begin{cases} \lambda \cdot ((h - h_{expected} + U(0,1)) & if\ h > h_{expected} \\ \dfrac{\lambda}{h_{expected} - h + 1} \cdot U(0,1) & if\ h \le h_{expected} \end{cases}. \quad (3)$$

As in the case of (1), delays computed according to (3) ensure that those nodes that are closer to the destination than the sender forward their packets before those that are not. Additionally, delays generated by (3) order the responses of those nodes that are at no closer to the destination than the sender according to their distance from the destination. Hence, no packet will travel a longer than necessary route, even when no node closer to the destination than the sender is available.

Second, nodes' transmission back-off delays are measured in transmission *slots*, instead of continuous time. In SHR, this is achieved by having a receiving node further determine its transmission back-off delay by the following equation:

$$d_{backoff\_slotted} = \left\lfloor \frac{d_{backoff}}{ts} \right\rfloor \cdot ts\ , \quad (4)$$

where $d_{backoff}$ is defined by (3) and *ts* is the width of the slot to be used. Later in Section 5, we present a theoretical analysis showing that the probability of packet collision induced by SHR with slotted delay is lower than that of SSR with continuous delay and we also derive an optimal value for *ts* used in (4).

3.2. Route Repair

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 no neighbors that are closer to the destination than itself. In this case, depending on the network topology, corrective action must be taken to reroute packets along the remaining shortest route. SHR achieves this by adjusting the hop distance of the sending node. The following is the set of rules for actions of senders and receivers that implement route repair using DATA and HELLO packets. This first set of rules only applies to the propagation of DATA packets, which excludes any packets used for the route setup phase. Senders behave according to the following rules:

1) Before transmitting an unseen packet for the first time, set a special *resend* bit in the packet header to 0, set a timeout timer, then transmit the packet;
2) If no responses other than an explicit ACK from the previous sender is detected before the timeout timer expires, then resend the packet with the resend bit set to 1;
3) Otherwise, if there is a response, send an explicit ACK packet, and if the expected hop count of the responder

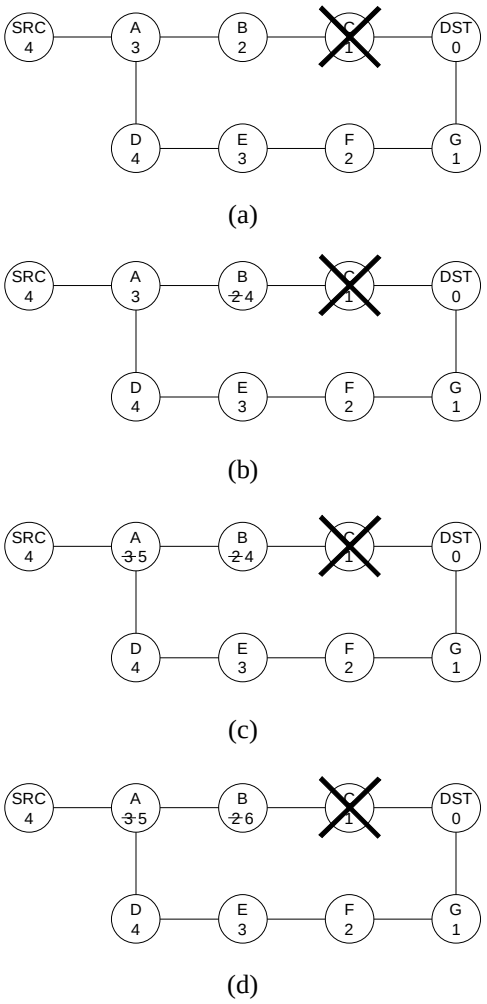

Figure 2. SHR route repair scenario.

is not one less than the current hop distance, change the hop distance corresponding to the packet's destination to be one more than the expected distance (the number of packets required to affect a change may be tuned; see Section 6.2 for a discussion);
4) If no response to a resent packet is detected, signal a higher protocol layer to take further action (the sender has become isolated and no forwarding is possible);
5) Upon receiving a response to a resent packet, change the hop distance for the packet's destination to 2 plus the maximum of the current hop distance and the expected hop distance from the response packet, then transmit an explicit ACK.

Receivers behave according to the following rules:

1) If the packet has its resend bit set to 0 and the packet for a particular destination was previously received (easily determined by use of sequence numbers), do nothing;
2) Otherwise, proceed as normal (i.e., determine the transmission back-off delay using Equations (3-4).

The result of the above rules is illustrated in Figure 2. Suppose node C has left the topology and node B has a packet to transmit as shown in Figure 2(a). Node A will respond and eventually cause node B's hop distance to increase to 4 as shown in Figure 2(b). In this scenario, node B will again respond to node A's broadcast, causing node A to increase its hop distance to 5 as shown in Figure 2(c). Node A will then again respond to node B's broadcast, eventually causing node B to increase its hop distance to 6 as shown in Figure 2(d). At this point, the packet, and all further packets, will travel along the new path to the destination. Note that for this scheme to be effective, as many nodes as possible must know their distances to the destination. We explain later in Section 4 how we address this concern in the route request-reply phase.

SHR also attempts to readjust nodes' hop distances in the case that the departing node returns to the network topology (a regular scenario for energy-efficient topology control algorithms). This is beneficial since it is assumed that original hop distances provide the shortest routes to the destination. When a node returns to the network (e.g., wakes up or repairs a fault), SHR readjusts nodes' hop distances using HELLO packets. A returning node broadcasts a HELLO packet with its last known hop distance to the destination for each known destination. Upon receiving a HELLO packet, a node reacts according to the following rules:

1) If the node's hop distance to the destination is less than or equal to the sender's hop distance plus 1, do nothing;
2) Otherwise, change hop distance to the sender's hop distance plus 1 and broadcast a new HELLO packet with the node's new hop distance to the destination.

We leave it to the reader to use the scenario in Figure 2 to see how this scheme re-establishes the original route.

4. The SHR Protocol

Here, we discuss how the algorithm described in Section 3 is used to build the SHR protocol. The primary data structure used by SHR is a cost table, an entry of which consists of the following items:

1) The identities (ID's) of a source and destination of the flow;
2) The sequence number of the last packet observed from the source node;
3) The hop distance from the destination to the current node.

The SHR protocol itself consists of three phases: *destination request*, *destination reply*, and *data transmission*.

4.1. Destination Request Phase

When a source node wants to send DATA packets to a destination for which there is no known distance, it first floods a *destination request (DREQ)* packet into the network. Each DREQ packet contains the following items:

1) The source node's ID;
2) The source node's sequence number which distinguishes the packet from other DREQ packets originating from the same source;
3) The destination node's ID;
4) The actual hop count which describes the number of hops that the packet has traveled from the source to the current recipient node.

The source initializes the actual hop count field to 1. After transmitting the DREQ packet, the source increments its sequence number by one.

If an intermediate node receives a DREQ packet from a source for which it has no cost table entry, it will create a new table entry using the packet's source ID, sequence number, and actual hop count fields. Otherwise, if a table entry for the source exists, the node will update the table entry's sequence number and hop count either if (a) the DREQ packet's sequence number is greater than that in the table entry (indicating the establishment of a fresher route), or (b) the DREQ packet's actual hop count is lesser than that the one stored in the table entry. The intermediate node will then increment the actual hop count field by 1 and attempt to rebroadcast the packet after some random back-off delay to avoid collisions with neighboring nodes. If the node receives another DREQ packet with the same source ID and sequence number before its transmission back-off delay expires, it simply cancels the packet transmission. When the destination receives the DREQ packet, it will not rebroadcast it. This phase results in all nodes knowing their hop distance from the source.

4.2. Destination Reply Phase

Upon receiving the DREQ packet, the destination will broadcast a *destination reply (DREP)* packet. The DREP packet contains the same items as the DREQ. The destination initializes the DREP's actual hop count field to 1 and increments its own sequence number by 1 after broadcasting the DREP packet.

At this point all nodes know their hop count from the source node. We assume that links are symmetric, so the hop distance *from* a target is a good measure of the hop distance *to* a target. When a node receives a DREP packet, it will update its cost table in the same manner as in the destination request phase, only this time the distance to the destination will be recorded using the actual hop count field.

SHR forwards DREP packets differently than does SSR. In SSR, the DREP packets are forwarded to the source node via the shortest route. As a result, only the nodes that are within reception range of nodes along the shortest path will learn of their distances to the destination node. Proper operation of SHR's route repair routine requires that nodes beyond the shortest route also know their distance to the destination node. For this reason, if a node receives a DREP packet and updates its cost table, it will increase the packet's actual hop count by one and forward the packet after a random delay. Upon receiving the DREP packet, the source knows that the destination exists and will broadcast an explicit ACK after a random delay (again, to avoid packet collisions), so that other nearby nodes can cancel their DREP transmissions and prevent redundant packet transmissions.

4.3. Data Transmission Phase

Upon receiving the DREP packet, the source can now start sending DATA packets to the destination. Since now there are nodes that know their distances from the destination, the SHR algorithm described in Section 3 can naturally be used to transmit each DATA packet to the destination.

5. Analysis

A fundamental improvement of the SHR algorithm over SSR is the organization of receiving nodes' responses into a slotted time space instead of a continuous one. Here, we show the magnitude of this improvement by analyzing and comparing each algorithm's probability of creating collisions among responding receivers. Also, for SHR, we determine the optimal value for the slot length, *ts*.

5.1 Comparison of Collision Probabilities

For the entire analysis in this section, we define a time unit to be equal to the transmission time of a packet. First, we make some preliminary remarks regarding SHR. For nodes with different hop counts to the destination, Eq. (3) yields back-off delays in disjoint intervals. Nodes closer to the destination than the sender all respond in $[0, \lambda)$ interval. Nodes that differ from the sender by k=0,1,… hops to the destination respond in the interval $[(k+1)\lambda, (k+2)\lambda)$. Hence, we need to consider here only slots within a single interval of size 3λ, which includes responses of all nodes closer by a hop, equal distance or farther by a hop from the destination than the sender. Let N be the number of slots over which a response can be transmitted, the time slot ts is $ts=3\lambda/N$, and, to avoid too many collisions, $\lambda>1$.

Now, we determine the probability of a receiver's transmission colliding with another one in SSR, which uses continuous time. Note that a receiver responds by starting its transmission at *any* time x in the range $[0, N\cdot ts)$ after receiving a packet. Hence, considering full and partial overlaps of the colliding transmission over the response time, the probability of collision is defined as follows:

$$P(continuous) = \int_0^1 \int_0^{x+1} \frac{dxdy}{N^2 ts^2} + \int_1^{N\cdot ts-1} \int_{x-1}^{x+1} \frac{dxdy}{N^2 ts^2} + \int_{N\cdot ts-1}^{N\cdot ts} \int_{x-1}^{N\cdot ts} \frac{dxdy}{N^2 ts^2} = \frac{2}{N\cdot ts} - \frac{1}{N^2 ts^2} \quad (5)$$

Since SHR uses slotted time, a receiver, after receiving a packet, picks a number x in the range $[0, N\cdot ts)$ and responds at time $k=\left\lfloor \frac{x}{ts} \right\rfloor ts$. Hence, a receiver will transmit at a slot time k=0, ts, $2\cdot ts$, …, $(N-1)\cdot ts$ with a probability of $\frac{1}{N}$. Letting $m = \left\lfloor \frac{1}{ts} \right\rfloor$, we have $m\cdot ts \le 1 < \lambda = \frac{N}{3} ts$, so $N>3m$, and therefore the collision probability when the node responds at time slot k can be expressed as follows:

$$P(slotted) = \sum_{k=0}^{m} \frac{k+1+m}{N^2} + \sum_{k=m+1}^{N-2-m} \frac{2m+1}{N^2} + \sum_{k=N-1-m}^{N-1} \frac{m+N-k}{N^2} = \frac{4m+1}{N} + \frac{1.5m(m+3)}{N^2} \quad (6)$$

Note that if $ts \le 1$, then $m\cdot ts > \frac{m}{m+1} > \frac{1}{2}$; hence, $4m > \frac{2}{ts}$. Using this fact and comparing (5) and (6), we conclude that if $ts \le 1$, then $P(slotted) > P(continuous)$, so we select $N<3\lambda$, so $ts>1$. As a result, the collision is possible only if two receivers pick the same slot time to respond. Hence the collision probability for the slotted case simplifies to

$$P'(slotted) = \sum_{i=0}^{N-1} \frac{1}{N^2} = \frac{1}{N}. \quad (7)$$

Thus, the ratio between collision probabilities becomes

$$r_{collision} = \frac{P(continuous)}{P'(slotted)} = \frac{2}{ts} - \frac{1}{ts^2 \cdot N}. \quad (8)$$

From the above equation, we can conclude that the only beneficial values of ts are within the interval (1,2) and the closer to 1 (i.e., the maximum transmission time) the time slot is, the better the ratio is. Clearly then, the best value of the number slots N is $N=\lfloor 3\lambda \rfloor$ and increasing λ enables increasing the number of slots N and thus improves the ratio and decreases the probability of collision in the slotted case.

A receiver's transmission can also collide with a transmission of a node that is attempting to forward an unrelated packet from another sender; we call such a collision an external one. Based on the above analysis, we only need to consider collision probabilities for $1<ts<2$. Hence, let $ts=1+e$, where $0<e<1$. We assume that the external transmission can start with equal probability density, p_e, at any time and lasts one time unit, so the collision probability can be expressed as:

$$P(slotted\_external) = \int_0^e \frac{p_e}{N} dx + \int_e^1 \frac{2p_e}{N} dx + \int_1^{ts} \frac{p_e}{N} dx = \frac{2}{N} p_e \cdot \quad (9)$$

In continuous case, the collision probability is defined as follows:

$$P(continuous\ \_external) = \int_0^{ts} \frac{2 \cdot p_e}{N} dx = \frac{2+2 \cdot e}{N} p_e \cdot \quad (10)$$

Comparing these two equations, it is clear that the probability of collision in slotted case is lower than in continuous one with the ratio of the two reaching ½ for *ts* close to 2. Hence, the final conclusion is that the best value of the time slot is between the maximum of the transmission time (including any additional time arising from the transmission delay of the response) and double of the minimum transmission time. SHR halves the probabilities of the internal or the external collisions at each extreme of this interval, and offers nearly 50% improvement for both near the middle of this interval.

6. Performance Evaluation

Here, we compare the performance of SHR with and without route repair (labeled SHR-RR and SHR in the plots respectively), SSR, and AODV under various network conditions and for different values of λ (where applicable). We include AODV in our comparisons because it is a well-accepted and well-documented wireless routing protocol whose non-opportunistic packet forwarding behavior is very similar to other traditional protocols mentioned earlier. Hence, it serves as a good basis of comparison to highlight SHR's strengths. The performance results were produced using the SENSE wireless network simulator [12].

The base configuration consists of a 2000 x 2000 m^2 terrain populated with 500 nodes placed randomly in this terrain. Each node is non-mobile and has a nominal transmission range of 250 m. The wireless medium was simulated with the free space propagation model [13] and

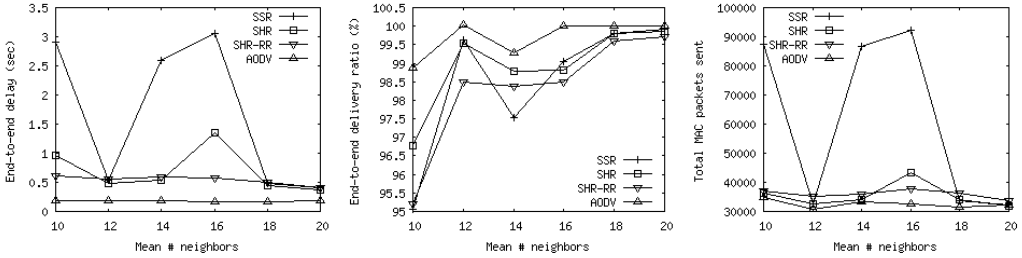

Figure 3. Protocols' performance based on the mean number of neighbors per node (network density).

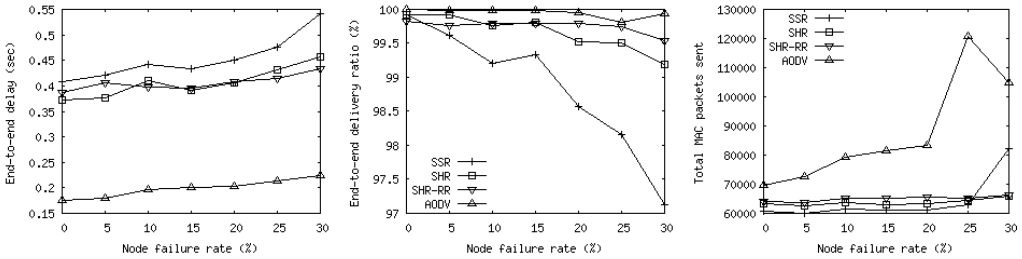

Figure 4. Protocols' performance based on nodes' permanent failure rates.

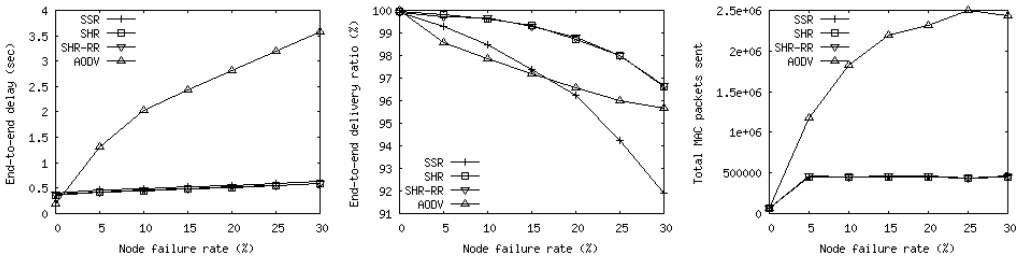

Figure 5. Protocols' performance based on nodes' transient failure rates.

the radio modeled operation at 914 MHz with 1 Mbits/sec bandwidth. The simulated WSN application sends packets with uniformly distributed size around the mean of 1000 bytes and at uniformly distributed intervals with a mean of 40 sec. MAC broadcast was used in which a node senses the carrier and broadcasts only if not other transmissions are detected. Each simulation was executed either eight or ten times with different random number seeds with the total simulation time of 3,000 sec. for all cases except transient and permanent failures that ran double of that time. The seeds varied between simulation sets.

We performed six sets of simulations, each comparing the protocols' performance against changes in one test parameter. The following test parameters were used: (i) the mean number of neighbors per node; (ii) the rate of permanent node failures; (iii) the rate of transient node failures; (iv) the number of source-destination pairs; (v) the number of sources communicating with a single destination; (vi) the value of λ. The slot width used for both SHR-RR and SHR is 12 ms.

6.1. Effect of Node Density

This test varied the network density by changing the mean number of neighbors per node from 10 to 20 nodes. If there are more neighbors, a packet can generally increases the physical distance traveled at each hop. However, the probability of collisions also increases as more neighbors compete to forward a packet. Figure 3 shows that SHR-RR's delay decreases as network density increases, thanks to the first effect mentioned above, a packet reaches its destination quicker. The data for SSR and SHR do not follow a similar trend. For some points on the plot, as the mean number of neighbors increases, SSR's delay increases enormously before eventually decreasing again. We attribute the high delays to packet collisions that cause SSR to forward packets along longer paths than actually available (see Section 2 for an explanation). SHR exhibits similar behavior, though not as pronounced. This shows a clear benefit of using slotted transmission back-off delays as opposed to continuous ones. SHR's route repair routine further improves upon the protocol's performance, granting it ideal behavior. Figure 3 also shows that the mean total number of MAC packets sent is proportional to mean delay. Since more packets are needed when collisions occur, this further supports above interpretation of the results. AODV has the lowest overall delay since it does not impose a delay at each hop to support opportunistic behavior.

Figure 3 also shows that on average, delivery ratio improves as density increases, as expected since more nodes are available to forward packets. Overall, AODV has the highest delivery ratio thanks to its unicast MAC layer that uses a request-to-send/clear-to-send based protocol (we show later that this solution is less effective in a fault-prone environment). On average, as density increases, SSR ranks best after AODV, followed in order by SHR and SHR-RR. This shows that while the simpler protocol may exhibit the better delivery ratio, it does so at a price of using more packets. We note that for a mean number of neighbors equaling 14, all tested protocols show a decrease in delivery ratio when collisions caused many packets to be lost, especially for SSR.

6.2. Effect of Node Failure Rate

We tested two node failure models, *transient* and *permanent*. 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. These simulation results 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 will be 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. SHR's route repair routine's sensitivity to failures may be tuned by adjusting the number of packets required to invoke route repair. By increasing this value, SHR can be successfully employed in a network with a high rate of transient failures. In our tests, two packets were required to invoke route repair. Since a packet transmission interval is 40 sec., then, a node failure lasting less than 80 sec. would not change the route from the source to the destination.

In both models, all nodes were active for the first 200 s of the simulation, or about the time it takes the source to transmit five packets. This allowed routes to stabilize. 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 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 s. The time spent in each mode was distributed exponentially about the mean value. This is a disruptive form of node failure since there is no attempt to coordinate it with the networking protocol.

The results with permanent node failures are shown in Figure 4. As the number of node failure increases, delay and delivery ratio generally increases and decreases, respectively. Both versions of SHR have a higher delivery ratio and lower delay than SSR. This again shows the benefit of the slotted delay formula. As the failure rate increases past 15%, SHR-RR's delivery ratio bests that of SHR, showing the clear benefit of having an additional route repair routine. However, the improved delivery ratio comes at the expense of transmitting about 7% more packets. Similar to Figure 3, AODV still exhibits the lowest delay and highest delivery ratios. However, it uses larger by the order of magnitude number of MAC packets to achieve this because the AODV's route repair algorithm initiates a new route request phase, causing a broadcast flood of packets from the point at which the route is severed in order

to find a new route. SHR's approach to route repair is clearly more local and efficient, as evidenced by the plots.

Figure 5 shows the results for the transient node failure model. In this test, both versions of SHR exhibit better performance characteristics than SSR. The nearly identical performance of both versions of SHR shows that the route repair routine is properly tuned for the network. AODV has the worst delay, significantly increasing along with the transient failure rate. Its delivery ratio is also lower than that of SHR and SHR-RR. Again, to achieve these results, AODV requires a significantly greater number of MAC packet transmissions than SHR and SHR-RR. We also note that while AODV's transmissions increase, SHR's and SHR-RR's remain nearly constant demonstrating that SHR's priority-driven opportunistic behavior is highly accommodative of potentially disruptive duty cycles and node failures.

In general, these two sets of failure tests highlight the fact that transient failures are much more disruptive to the network. This is evidenced by the delivery ratio that is several percents lower when the transient failure rate is 30% as compared to a permanent failure rate of 50%.

6.3. Effect of Network Traffic Volume

We performed two tests to understand an impact of the network traffic volume. The first test increased pairs of nodes that communicated with each other exclusively. The second test increased the number of source nodes communicating with a single destination node, a situation that is common in wireless sensor networks. The results of these tests are shown in Figures 6 and7, respectively.

On average, SSR, SHR, and SHR-RR all behaved in a similar manner. One notable difference is that implementation of the route repair routine has an overhead that requires sending additional packets. Even though none of the nodes fail, the occasional collisions will appear to the route repair routine as transient link failures, increasing the number of MAC packets sent. Delays generally increased and delivery ratios decreased with the growing traffic volume, although by very small if not negligible factor. Figure 6 shows that for the number of communicating source pairs equaling 18, SHR's delay markedly increases, and then returns to a normal trend. Inspecting the total MAC packet transmissions for this data point shows similar anomalous behavior, indicating a point where collisions caused packets to take much longer paths and subsequently incur larger delays. AODV exhibited marginally better

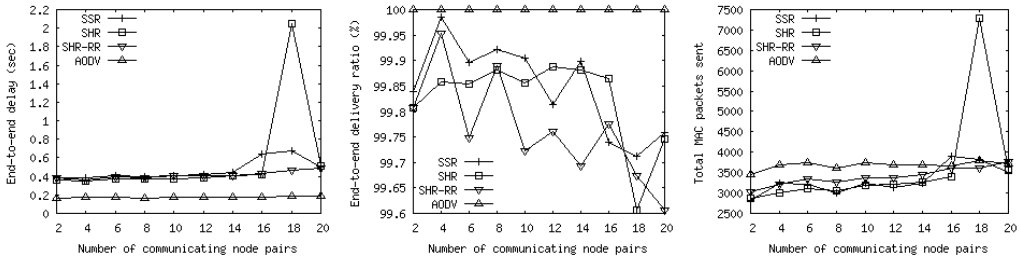

Figure 6. Protocols' performance based on the number of source-destination pairs.

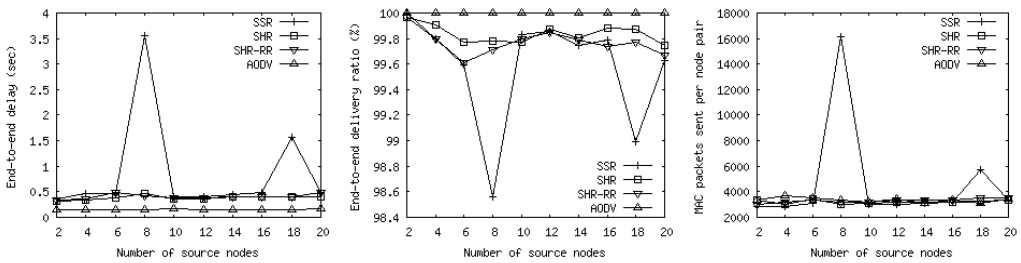

Figure 7. Protocols' performance based on the number of sources communicating with one destination.

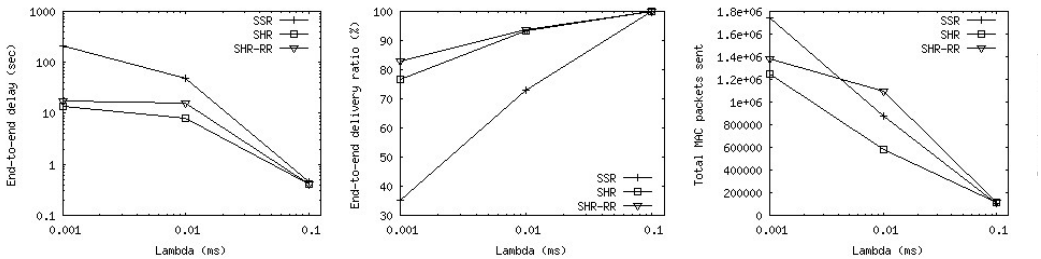

Figure 8. Protocols' performance based on the value of λ.

performance for delay and delivery ratio statistics. For an large number of communicating node pairs, SHR required more MAC packets than SSR or AODV, which is expected since the network topology is fixed and SHR is an extension of SSR that is designed to better handle dynamic networks.

6.4. Effect of Scaling Factor

The final test compared the performance of SSR, SHR, and SHR-RR as λ, the scaling factor, was increased. The results, shown in Figure 8, demonstrate that as λ decreases, there is degradation in all aspects of the protocols. The slotting mechanism of SHR provides a tolerance for small values of λ. When the value of λ is less than the slot time, all packets will be forwarded at the same time regardless of the election results. This will increase the number of MAC packets sent and decrease the delivery ratio due to collisions. These results confirm the conclusion of the analysis in Section 3 that the selection of λ and of the length of time slot *ts* must be made carefully in relation to the maximum transmission time of a packet.

7. Related Works

There are other protocols that, like SHR, route on the premise of avoiding neighbor state maintenance and letting receivers contend for forwarding packets. Two such protocols, GRAd [14] and GRAB [15], are similar to SHR in that they avoid the use of geographical location information. However, neither one uses a time slotting or a route repair. GRAB also uses a more aggressive fault-tolerance technique by allowing redundant packets to follow multiple paths to a destination. SHR forgoes this approach and relies strictly on its prioritized transmission back-off delay technique to enforce fault-tolerance.

Other opportunistic protocols rely on geographic location information to support routing decisions. For instance, BLR [16] 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, are allowed to contend to forward packets. A prioritized back-off delay scheme, similar to SHR, 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 [17] also employs a similar eligibility region with a prioritized back-off delay technique. However, GeRaF also 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 overheard than SHR's approach. IGF [18] 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 [19] and SIF [20].

SHR also draws similarities from the slotted version of the classic ALOHA random access protocol for wireless networks [21]. SHR and slotted ALOHA both use time slots in a similar manner to reduce and recover from collisions. However, slotted ALOHA is a MAC protocol, and hence makes no provisions for managing (and prioritizing) nodes' transmissions according any to routing-centric factors, e.g., the hop distance from a destination.

8. Conclusion and Future Works

In this paper, we have presented SHR, which naturally accommodates fault-prone network routing conditions. SHR's use of broadcast communication and prioritized transmission back-off delay with slotted time allows receiving nodes to make packet forwarding decisions based only on their hop distance from the destination. Additionally, SHR uses these features to provide seamless route repair.

We have identified several future directions for enhancing SHR. We started with evaluating SHR's performance on an actual Berkley mote-based sensor network and validated our simulations on a small 24-mote network [24] running TinyOS [25] (the *de facto* standard operating system for sensor motes). We plan to collect results on large networks, similar in size to the networks simulated here to evaluate SHR using a real-life radio signal under various environmental conditions (e.g., difference climates, indoors, outdoors, obstacles, etc.).

Next, we intend to extend SHR to support mobile nodes, which are characteristic of some WSN applications. While SHR may currently accommodate mobility, it is not yet explicitly optimized for it. Mobility will decrease the length of time over which cost tables remain valid. To retain SHR's autonomic behavior, we are researching how to efficiently update tables based on local observations of node movement.

Finally, we intend to make SHR more energy-efficient. Currently, SHR allows the random selection of preferred neighbors to which to forward packets. This helps balance network energy usage, but the behavior is not explicit. The limits of SHR's current random selection behavior become apparent when either there are nodes that consume more energy than others (e.g., due to excessive retransmissions 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 SHR'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 SHR can already accommodate topology changes, which could possibly be caused by energy-efficient topology control algorithms (e.g.,

GAF [22] and ESCORT [22]). However, the challenge with explicitly incorporating a topology control algorithm into SHR is making sure that the algorithm is not so aggressive that it negatively affects SHR'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 SHR.

9. Acknowledgments

The authors thank Gilbert G. Chen for his insight on this paper as well as his contributions in earlier research and Geoff Merrett at 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 and 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 [Online], 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 and 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. 2005. "Self-selective routing for wireless ad hoc networks." In *Proc. of IEEE Int. Conf. Wireless and Mobile Computing*, vol. 3, 57-65.

[11] 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.

[12] 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.

[13] T. S. Rappaport. 1996. *Wireless Communications: Principles and Practice*, Prentice Hall.

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

[15] 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).

[16] 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).

[17] M. Zori and 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.

[18] 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.

[19] 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*.

[20] 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*.

[21] N. Abramson. 1970. "The ALOHA system – another alternative for computer communications," in *Proc. Fall Joint Computing Conf., AFIPS Conference*, 37.

[22] Y. Xu; J. Heidemann; D. Estrin. 2001. "Geography-informed energy conservation for ad hoc routing." In *Proc. ACM MobiCom*, 70-84.

[23] 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.

[24] K. Wasilewski, J. Branch; M. Lisee; B.K. Szymanski. 2007. "Study in Resiliance of Wireless Sensor Networks." In *Proc. 4th Conf. on Networking, Springer-Verlag LNCS*, vol. 3420, 438-448. Air, Space and Underground Sensors, the conference of great importance and validity.

[25] TinyOS Community Forum. http://www.tinyos.net.