Conference on Unattended Ground, Sea, and Air Sensor Technologies and Applications,
SPIE Symposium on Defense & Security 2007

Self-Healing Routing: A Study in Efficiency and Resiliency of Data Delivery in Wireless Sensor Networks

Kamil Wasilewski[*], Joel W. Branch, Mark Lisee, Boleslaw K. Szymanski
Dept. of Computer Science, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY USA 12180

ABSTRACT

This paper presents the results of implementation of a novel protocol, Self-Healing Routing (SHR) for opportunistic multi-hop wireless communication, on MicaZ sensor motes. The protocol uses broadcast communication and a prioritized transmission back-off delay scheme to empower a receiving mote to use its hop distance from the destination to decide autonomously whether to forward a packet. When severed routes are encountered, the protocol dynamically and locally re-routes packets so they traverse the surviving shortest route.

We have implemented this protocol on a set of MicaZ motes as well as in the SENSE sensor network simulator and conducted field testing with different mote and network configurations. We also tested scenarios with the motes turned off (modeling permanent failures) and in simulation also temporarily off line (modeling transient failures). We compared SHR with two traditional protocols: MintRoute and AODV. The results, as shown by experimental measurement and simulations reported in the paper, demonstrate that Self-Healing Routing is an efficient fault-tolerant protocol that performs well even with spontaneous network topology changes.

Keywords: Wireless sensor networks, wireless communication, fault-tolerant routing, performance measurement

1. INTRODUCTION

1.1 Motivation and contributions

Increasing advances in hardware miniaturization and low-power wireless communication technology have supported the rapid proliferation of wireless sensor networks (WSNs) [1]. However, these networks present challenges for routing caused by routing faults (e.g., packet loss and delay) arising from unpredictably transient wireless links and malfunctioning or destroyed motes which are common in WSNs [2], [3] and limited power, often supplied by non-rechargeable batteries [4].

Traditional routing protocols often use routing tables that indicate to each mote the exact neighbor to which a packet should be sent in order to reach a specific destination. Prominent examples of such an approach include AODV [5], MintRoute [2], and Directed Diffusion [6]. This fundamental *unicast* routing approach intrinsically requires motes to actively maintain knowledge of who their neighbors are and what their neighbors' states are (e.g., *active*, *sleeping*, *destroyed*). Additional techniques may also be needed to judge the quality of all available links [7]. For a highly dynamic network such as a WSN, such an approach imposes high algorithmic overhead. Requiring a mote to constantly learn the details of its local connectivity and/or notify its neighbors of its chosen state in the midst of a dynamic environment requires a significant amount of additional radio activity not directly involved in transporting sensor data. Moreover, while routing messages to specific neighbors is adequate for ideal *static* network conditions, such an approach incurs additional delay in a dynamic environment as motes probe for new neighbors to forward packet to and reconfigure the network's topology.

The majority of WSN-related research activities have used network simulators such as ns-2 [8] and SENSE [9] to demonstrate the benefits of employing various routing protocols, often in comparison to other protocols. While simulation serves as an effective tool for verifying a protocol's correctness and making a well-informed prediction of a protocol's real life performance, it has limitations in emulating real WSN characteristics. Those include accurate modeling of wireless signal propagation behavior (and how it changes spatially, temporally, or with certain weather

[*] wasilk@cs.rpi.edu

conditions and terrains) and accurate modeling of inconsistent or erroneous device behavior (and how it affects a protocol's performance or even a device's rate of energy consumption). Furthermore, the collective experience of coding and debugging the protocol on hardware, configuring physical test scenarios, and reporting real world performance statistics and related artifacts is not possible.

In this paper, we address the above points by describing Self-Healing Routing (SHR), which represents our continuing research in fault-tolerant WSN routing, and presenting a detailed analysis of the protocol's performance using both simulation and a physical WSN platform. In SHR, instead of deciding where to forward a packet, a mote freely broadcasts it to all neighbors within its transmission range. Receivers then autonomously decide whether to forward the packet using only knowledge of their hop distances from the destination. We present two versions of SHR. The first, *SHR*, uses a prioritized transmission back-off delay scheme to ensure that a packet is forwarded by only *one* of the contending receivers. SHR also employs a local and efficient route repair mechanism. The second, *SHR-M*, which is more simplistic and liberal, essentially relaxes the requirement (avoiding the related overhead) that only one receiver can forward a copy of the same packet and does not use a sophisticated route repair mechanism. Both, SHR and SHR-M dynamically and locally determine the shortest routes, even in the context of spontaneous topology changes. In addition to presenting simulation results, we also present results and describe our experience from implementing SHR and SHR-M on a WSN using Crossbow® MicaZ sensor motes with the TinyOS development environment [10], [11]. This includes a detailed discussion comparing the simulated and actual performance of both protocols.

1.2 Related works

Like SHR and SHR-M, there are other protocols that route on the premise of avoiding neighbor state maintenance and letting receivers contend to forward packets. Two such protocols, GRAd [12] and GRAB [13], are similar to SHR in that they avoid the use of geographical location information. However, neither one provides for explicit route repair. GRAB also uses a more aggressive and complex fault-tolerance technique by actively enforcing the flow of redundant packets to follow multiple paths to a destination. Other opportunistic protocols rely on geographic location information to support routing decisions. For instance, BLR [14] uses location coordinates to allow only receivers in an "eligibility region," defined as a region in which all motes are closer to the destination than the sender and can overhear each others' transmission, are allowed to contend to forward packets. Similar to SHR, a prioritized back-off delay scheme ensures that the closest mote 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 links. GeRaF [15] 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 uses a request-to-send/clear-to-send (RTS/CTS) packet forwarding technique which imposes more packet forwarding overheard than SHR's approach. IGF [16] 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 motes, it will shift the eligibility region and try to find other receivers. Other similar location-based protocols include PSGR [17] and SIF [18].

The remainder of this paper is organized as follows. Section 2 describes the detailed behavior of SHR and SHR-M. Section 3 compares the performance of SHR and SHR-M to traditional unicast protocols on MicaZ motes while Section 4 provides this comparison based on simulation. Section 5 concludes the paper.

2. THE SHR AND SHR-M PROTOCOLS

Here, we summarize the SHR and SHR-M protocols, as more detailed description is provided in [19]. The primary data structure used by the protocols is a cost table, of which an entry consists of the following items: (i) the identity (ID) of a target mote, which is either a source or destination; (ii) the sequence number of the last packet observed from the target mote; and (iii) the hop distance from the target mote to the current mote.

The SHR and SHR-M protocols themselves consist of three phases: *destination request*, *destination reply*, and *data transmission*. The destination request and destination reply phases are identical for both protocols.

2.1 Destination request phase

When a source mote wants to send DATA packets to a destination for which there is no known route, it first floods a *destination request (DREQ)* packet into the network. Each DREQ packet contains the following items: (i) the source mote's ID; (ii) the source mote's sequence number which distinguishes the packet from other DREQ packets originating from the same source; (iii) the destination mote's ID; and (iv) the actual hop count which describes the number of hops the packet has traveled from the source to the current recipient mote.

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 mote 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 mote 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 stored in the table entry. The intermediate mote 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 motes. If the mote 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 motes knowing their hop distance from the source.

2.2 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 with the addition of the expected distance to the source. The destination initializes the DREP's actual hop count field to 1, the expected hop count field to the value from the cost table, and increments its own sequence number by 1 after broadcasting the DREP packet.

At this point all motes know their hop count from the source mote. 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 mote 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.

Proper operation of SHR's route repair routine requires that motes beyond the shortest route also know their distance to the destination mote. For this reason, if a mote 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.

2.3 Data transmission phase

Upon receiving the DREP packet, the source can now start sending DATA packets to the destination.

The SHR-M protocol. The data transmission phase of the SHR-M protocol uses a single packet type, DATA, that contains the same items as the DREP packet with the addition of a data payload. The source initializes the actual hop count field to 1 and the expected hop count field to the destination's value from the cost table. After transmitting the DATA packet, the source increments its sequence number by one. Neighbors that are not closer to the destination ignore this packet and all future packets with the same sequence number. A mote determines if it is closer to the destination than the packet's sender by comparing the value of the destination's entry in the cost table to the packet's expected hop. Neighbors that are closer record the expected hop from the packet and start a timer that is uniformly distributed between 0 and λ. The value λ is a scaling factor that reduces the probability that the motes' responses will collide. The cost of increasing λ is a corresponding increase in the delay to deliver a packet to its destination. If the mote receives another copy of the packet, it compares the packet's expected hop to the recorded expected hop. If the new packet is not closer to the destination, it is ignored. If it is closer, the mote cancels its timer and ignores future packets with this sequence number. When the mote's timer expires, the mote increments the packet's actual hop, sets the expected hop to the value of the destination's cost table entry and transmits the packet. This mote will ignore all future packets with this sequence number. When the destination receives the DATA packet, it forwards the data to the application but does not transmit any packets.

The exclusion of *acknowledgement (ACK)* packets cuts the number of packet transmitted in half in the ideal case. But, under less than ideal conditions, it is possible that the DATA packet could be dropped before reaching the destination or that multiple copies of the packet are delivered to the destination. A DATA packet can be dropped when the motes' cost tables become stale and incorrectly imply that a route exists. The traffic stream may fork when a mote has neighbors that cannot communicate directly to each other, resulting in duplicate DATA packets arriving at the destination. ACK packets address this situation and are one of the significant differences between SHR-M and SHR.

The SHR protocol. The data transmission phase of the SHR protocol uses two packet types, DATA and ACK packets. The DATA packets contain the same fields as the SHR-M DATA packets with the addition of a maximum hop field. This field indicates the maximum number of times that a packet may be forwarded before it must be discarded. This field is initialized to a value that is larger than the expected hop. The ratio between the maximum and expected hop values is a tuning parameter. A large ratio will allow the route repair mechanism to recover from more extreme breaks in the network topology but may cause the transmission of an excessive number of packets. The ACK packets contain only the source address, destination address and sequence number. These fields are initialized with the values from the fields in the corresponding DATA packet. In SHR, each mote maintains an *IgnoreCount* for each flow, or source-destination address pair, that it has seen. This value indicates if the mote should participate in a forwarding election for that flow and is further described below.

When the source transmits the DATA packet, only neighbors that are closer to the destination than the sender will respond. Unlike SHR-M, such a neighbor may respond even to packets with the already seen sequence number. If *IgnoreCount* is not zero, then it is decremented and the mote will ignore this and all future packets with this sequence number. Otherwise, the mote selects a transmission back-off time as described for SHR-M. If, during this time, the mote receives a DATA packet from a mote that is closer to the destination, it cancels the forwarding of the DATA packet but lets the timer continue. If it receives an ACK or a second DATA packet, it sets *IgnoreCount* to a non-zero value and ignores all future packets with this sequence number. The maximum value of *IgnoreCount* is a tuning parameter.

When the transmission back-off time expires, the mote sets the actual and expected hop fields, as described for SHR-M, and transmits the packet. Unlike SHR-M, the mote selects another random period with end-time in the range 1.25λ to 1.75λ during which it monitors the carrier to determine if the packet has been forwarded. Ideally, the mote should hear only a single DATA packet that was transmitted by a mote that is closer to the destination. If the mote detects a second DATA packet, then there is the situation in which two or more neighbors are out of each other's transmission range. In this case, the mote sends an ACK packet. The ACK will reduce the number of neighbors that participate in the election for the next DATA packet in this flow.

If the mote determines that the packet was not forwarded, it retransmits the packet and continues to listen for another 1.25λ to 1.75λ. If, during this time, the mote receives an ACK or DATA packet, the mote cancels the timer and will ignore all future packets with this sequence number. If the timer expires, the mote undertakes route repair by adding two to the value of the destination's entry in the cost table. If the new distance plus the packet's actual hop field is less than the maximum hop field, the mote will transmit the packet with the new distance as the expected hop. After transmitting the packet, the mote will ignore all future packets with this sequence number.

When the destination receives the DATA packet it transmits an ACK packet and starts a timer for 10λ. Any DATA packets received during this time period cause the destination to send another ACK packet. After the time period, the destination ignores all packets with this sequence number.

3. EXPERIMENTAL RESULTS

3.1 Experiment setting

All protocols were tested three times for a particular topology and the average of these runs was recorded since mote and antenna orientation have a significant impact on radio performance [4]. Each protocol’s DATA packets were 29 bytes long to maintain the same transmission time for each DATA packet. Atmospheric conditions also have a significant impact on radio performance [4]. To minimize this effect, for a given topology, testing was done on all protocols in the span of a few hours. TinyOS version 1.1.7 was used but the MicaZ CC2400 radio library was updated to allow for the use of the time stamping interface as described in [20]. B-MAC with acknowledgements disabled provides link layer functionality. DATA packets are sent for 12.5min at a rate of 5sec/packet.

The testing environment was in the atrium of the Rensselaer Polytechnic Institute Biotechnology Building with a stone floor and on an outdoor artificial turf field on the Rensselaer campus. Testing was done at night when foot traffic is low. To limit the range of transmission, the motes were placed directly on the ground. To determine a distance which provides a reliable delivery rate but minimizes the possibility of a mote transmitting further than to adjacent motes, first pairs of motes were placed at varied distance and the delivery rate recorded. Then, the distance that provided a delivery rate of approximately 90% was used. Experimentally it was determined that at 0dBm on the artificial turf a distance of 5m provided a reliable connection. In the indoor environment, where space is more limited, the radio power was reduced to -21dBm and a distance of 2m provided a reliable delivery rate. However, as shown in [2], it is still likely that some reliable long distance links will form.

MicaZ motes were numbered and placed in predetermined locations. The destination (or base) mote was placed on the MIB510 programming board powered by an AC power supply, which was attached to a laptop to collect data. With the exception of the base mote, all motes are placed right side up parallel to the base station with the antenna side pointing away from the base station and the antenna itself oriented perpendicular and away from the floor.

3.2 Evaluation metrics

End to end Delay is the amount of time required for a packet to travel across the network from the source to the destination. The average delay is calculated to compare the tested protocols. To measure it, we used a send and receive time stamp [20]. First, all motes were positioned in predetermined locations except for the source which was placed next to the destination. The source then sent 10 packets with a send time stamp and the destination stamped its receive time. The source is then returned to its predetermined place and data transmission begins. Since the internal clocks of the mote CPUs can drift at a predictable rate, as long as environmental factors remain stable, up to 40µs/s it is necessary to compensate for this skew. Figure 1 shows the growing clock skew, which remains linear and predictable given the same working conditions. Skew compensation with linear regression [21] is used offline to correct end-to-end delay.

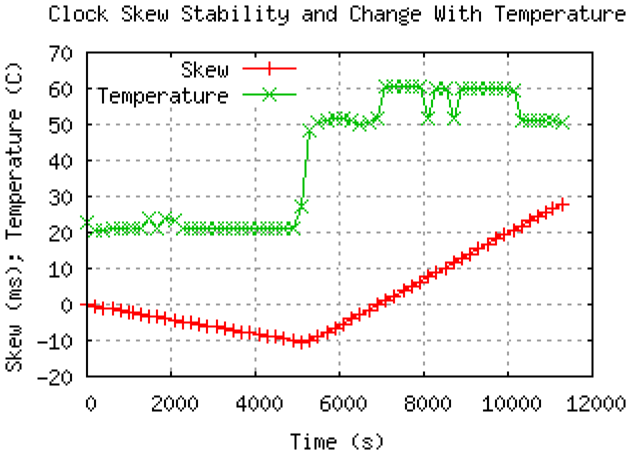

Fig. 1. The predictability of the clock skew of a MicaZ mote is shown here. When the temperature surrounding the mote changes the clock skew also changes as the internal oscillator can be affected by environmental conditions. However, as one can see the skew remains stable in a given environment.

Delivery rate is computed from the total unique packets arriving at the base mote divided by the total unique packets sent by the source.

MAC packet count is the number of packets sent and received over the radio of the mote from which we can determine the ratio of received to sent packets and the average number of MAC packets per hop. Counting is done by an interface to the MicaZ CC2420 radio. The implemented interface increments a send and receive counter whenever a packet is sent or received over the radio with the option to reset the counters at any time from the routing layer. The MAC packet count starts when a route from the source to the destination is formed. In the case of SHR this is after the DREQ/DREP phase. For MintRoute the network is given time to form and stabilize a path from the source to the destination, at which time the counters are reset and the official count begins. The count ends when a special broadcast is sent from the source to all motes. The motes then broadcast their counts to the base station.

3.3 Testing scenario

The routing protocols evaluated on MicaZ motes include: (i) SHR-M and (ii) SHR, both with λ=22ms and cost table of size 24; and (iii) MintRoute in which link-quality estimates (rather than minimum hop count) are used to select a parent that minimizes the expected number of transmissions to reach the root of the network. MintRoute v1.7 with the window mean exponentially weighted moving average (WMEWMA) link estimator was used. All MintRoute settings were left default. The Surge application was used to send a DATA packet every 5 seconds from the source mote.

To test the protocols and their behavior in certain environments we have created different topologies each testing a specific aspect of their behavior.

Double line, Figure 2(a), allows for two motes per hop to overhear a broadcast increasing the probability that the packet will be forwarded. This topology tests how SHR can use the extra motes to repair a route if link connectivity changes over time.

Route Repair, Figure 2(b), offers three unequal disjoint paths: a short, medium and long path. With this topology we test the repair characteristics of the protocols. Desirable behavior is to have the shortest recovery time possible when a mote is destroyed and choose the next shortest path that offers adequate connectivity. During testing we block mote 12 and 13 from the network by placing a metal container over the motes after the first 5 minutes of the test

Multiple equal paths, Figure 2(c), provide redundancy which in a broadcast network results in higher reliability. The source and destination have connectivity with all first hop motes allowing for four different paths a packet can take to arrive at the destination.

Multiple sources grid, Figure 2(d), shows the effect that multiple sources can have on the network. Multiple sources increase traffic density and rate. Traffic is sent in a round robin fashion from the sources. This grid provides redundancy increasing the likelihood of packet delivery in a broadcast network.

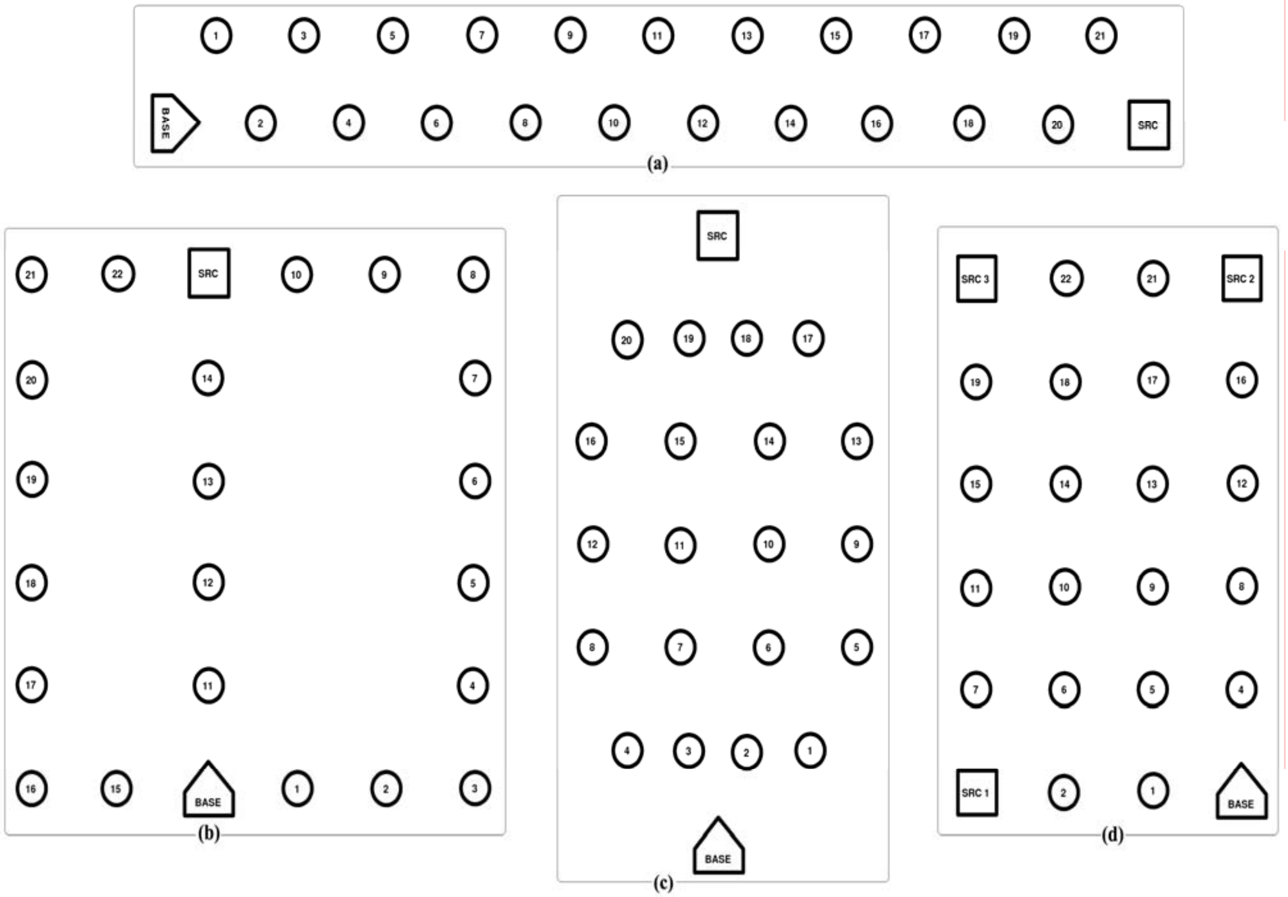

Fig. 2. Nodes only have connectivity with close neighbors but there exits a probability of long distance connections. The base indicates the direction in which all motes are oriented. (a) Double line topology. (b) Route repair topology with unequal paths. No communication exists between paths. (c) Multiple equal paths. (d) Multiple sources topology.

3.4 Results

SHR-M performs poorly in most topologies because of a limitation of the route discovery phase. The base station has more power than other motes as it is powered by an AC adapter, so it is able to transmit a DREP packet further then other motes. This forces all motes that overhear the DREP packet to believe they are 1 hop away from the destination. When the source sends DATA packets they are only forwarded to the edge of the group of motes that overheard the base station's DREP broadcast. In this group, the mote closest to the source broadcasts the DATA packet but is too far for the packet to be received at the destination. Such packets are not forwarded by other motes since the expected hop count is 1. To test this explanation, the antenna of the base mote was angled toward the ground, reducing its transmission range, causing the DREP to be heard by only adjacent motes. As expected the delivery rate increased.

Table 1. Experimental results for double line and route repair topologies.

	Double line Topology			***Route Repair Topology***		
	SHR-M	***SHR***	***MINTROUTE***	***SHR-M***	***SHR***	***MINTROUTE***
Packets sent	969	1,468	19,252	468	3,368	12,882
Packets received	3,960	6,772	117,213	1,232	8,563	41,050
Packet ratio (receive/send)	4.10	4.61	6.08	2.63	2.54	3.19
Delivery rate	6.67%	31.11%	86.44%	24%	78%	79.33%
End to end delay	56.82 ms	85.14 ms	32.02 ms	48.50 ms	128.16 ms	24.54 ms
Average hop count	6.26	7.18	6.99	5	6.10	5.99
Route setup time	25 sec	30 sec	180 sec	5 sec	5 sec	190 sec

SHR also suffers from a poor delivery rate during the beginning of the data transmission period but gradually improves delivery as shown in Figure 3. The initial low delivery is due to the unbalanced distribution of hop count away from the base station. This forces the protocol to perform route repair increasing the delay of the packets arriving. Once a reliable route has been discovered, the end-to-end delay decreases significantly even though the hop count increases. The path becomes reliable and requires fewer retransmissions to deliver the packet.

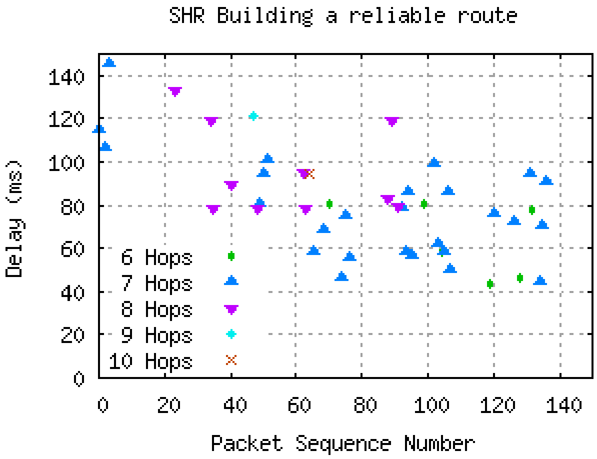

Fig. 3. SHR recovery and building of a reliable route in the double line topology. The delay decreases while the hop count increases gradually when consecutive repair phases establish reliable path.

MintRoute offers a high delivery rate compared to SHR-M and SHR but at the cost of a greatly increased MAC packet count, Table 1. For a 2.7 times increased delivery rate, MintRoute requires 16 times more packets. A simple 4-fold packet replication scheme in SHR will increase its delivery rate above MintRoute and still leave it with 4 times fewer packets sent. The end-to-end delay of MintRoute is lower than either SHR or SHR-M since it does not perform back-off delays. However, the setup time required by MintRoute is much greater as it must form its tree while SHR depends only on the delivery of its DREQ/DREP packets.

SHR-M is not able to repair a broken route. The protocol depends entirely on the initial DREQ/DREP phase to provide route information. If this information changes during the life of the network, SHR-M will not compensate, so if a mote on a path is removed, SHR-M will fail to deliver any packets, see Figure 4(a). In contrast, SHR can repair a broken route and find the next shortest and reliable path. As seen in Figure 4(b), SHR quickly finds the next shortest path and delivers data to the destination. Even the longest path of the topology will be used to compensate for dropped packets. Hence the removal of motes is not detrimental to SHR's performance. MintRoute recovers from the broken shortest path but required 150 seconds to do so, see Figure 4(c). The destruction of motes can be devastating to MintRoute, making it inadequate for a situation where motes can be compromised.

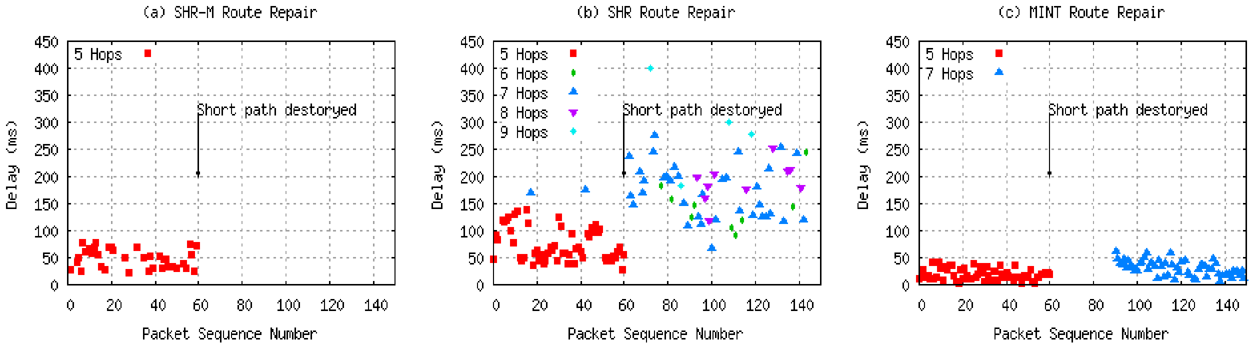

Fig. 4. Delivery, delay and hop count of protocols. The shortest path was destroyed after packet with sequence number 60. (a) SHR-M offers no route repair so it drops delivery rate to 0 after the shortest path is blocked. (b) SHR responds with recovery and increase in hop count when the shortest path is destroyed. The longest path is occasionally used when the medium path is under repair. (c) MintRoute recovers from the failure but after a much longer time reducing its delivery rate significantly.

SHR-M performs better in a high density environment thanks to the presence of additional motes able to forward packets, but SHR's delivery rate drops because of a large number of collisions resulting from the small λ value used. Increasing λ creates longer back-off delays but reduces the risk of collisions. The performance of MintRoute on this topology is also lower. A high mote density causes instability in the protocol by imposing frequent restructure of its routing tree. As seen in Table 3, the delivery rate of SHR-M approaches that of MintRoute but it uses 15 times fewer MAC packets.

Table 2. Experimental results for multiple equal paths and multiple sources grid topologies, both are of high density.

	Multiple equal paths Topology			*Multiple sources grid Topology*		
	SHR-M	***SHR***	***MINTROUTE***	***SHR-M***	***SHR***	***MINTROUTE***
Packets sent	596	1,794	17,365	1,474	4,239	19,864
Packets received	1745	5,587	97,246	5,937	16,689	98,343
Packet ratio (receive/send)	2.27	3.11	5.60	4.03	3.94	4.95
Delivery rate	17.7%	30.7%	72.67%	54.9%	36.6%	70.11
End to end delay	16.3 ms	60.4 ms	14.18 ms	21.6 ms	58.7 ms	16.34 ms
Average hop count	3.15	3.85	4	3.49	3.63	3.41
Route setup time	5 sec	5 sec	200 sec	10 sec	8.33 sec	200 sec

4. SIMULATION OF EXPERIMENTAL TOPOLOGIES

In addition to the real network tests, we used the SENSE network simulator [9] to simulate the SHR-M and SHR protocols on the same topologies. Each topology was simulated with two slightly different conditions. In the first simulation, when a mote transmitted a packet, it had a 90% chance of being successfully delivered to the next mote. Since this does not accurately reflect the observation that some packets were skipping over motes, the simulation was run again with a packet having a 5% chance of being successfully delivered to the neighbor's neighbor. The probabilities of delivery on the single and double hop links were independent of each other. The results are shown in Tables 3 through 6. Each simulation result in the table represents an average of ten runs using different random number seeds.

The two-hop link reduced SHR's delivery rate by about 10%. SHR's automatic retry and route repair work together to alleviate the dynamic nature of the topology. SHR-M, which does not retransmit packets and lacks route repair, responds very poorly to changes in the topology. When a packet is forwarded along one of the two hop links, the mote that sent the packet will think that it's closer to the destination and lower the value in the cost table. Future packets that use the one hop link will be dropped because the receiving mote is not closer than the sending mote.

The two mesh topologies, Tables 4 and 5, show that SENSE is optimistic with the packets that it considers successfully delivered in highly connected topologies. Both tests of SHR and the single hop test of SHR-M have delivery rates that are much higher than the experimental results because the simulated links are based on a connectivity matrix and do not consider the signal attenuation.

Table 3. Experimental and simulation results for the double line topology.

		SHR-M			***SHR***	
	Experimental	***Simulation***	***Sim. with skip***	***Experimental***	***Simulation***	***Sim. with skip***
Packets sent	969	1,046	379	1,468	2,880	2,919
Packets received	3,960	3,495	1,092	6,772	9,636	10,078
Packet ratio (receive/send)	4.09	3.34	2.89	4.61	3.35	3.45
Delivery rate	6.67%	31.6%	0.40%	31.1%	67.5%	54.9%
End to end delay	56.8 ms	119.2 ms	56.7 ms	85.1 ms	280.8 ms	342.6 ms
Average hop count	6.26	11.04	10.4	7.18	12.5	12.1

Table 4. Experimental and simulation results for the multiple equal paths topology with one source.

		SHR-M			***SHR***	
	Experimental	***Simulation***	***Sim. with skip***	***Experimental***	***Simulation***	***Sim. with skip***
Packets sent	596	1,542	362	1,794	2,292	2,378
Packets received	1745	7,750	1,676	5,587	11,682	12,956
Packet ratio (receive/send)	2.27	5.03	4.63	3.11	5.10	5.45
Delivery rate	17.7%	94.7%	4.53%	30.7%	89.8%	73.1%
End to end delay	16.3 ms	38.5 ms	31.1 ms	60.4 ms	75.6 ms	112.8 ms
Average hop count	3.15	6.00	4.11	3.85	6.06	5.54

Table 5. Experimental and simulation results for the multiple sources grid topology.

		SHR-M			***SHR***	
	Experimental	***Simulation***	***Sim. with skip***	***Experimental***	***Simulation***	***Sim. with skip***
Packets sent	1,474	2,118	583	4,239	5,848	6,879
Packets received	5,937	11,395	2,165	16,689	29,174	36,445
Packet ratio (receive/send)	4.03	5.38	3.71	3.94	4.99	5.30
Delivery rate	54.9%	85.2%	6.58%	36.6%	84.4%	75.3%
End to end delay	21.6 ms	31.6 ms	16.7 ms	58.7 ms	66.2 ms	82.2 ms
Average hop count	3.49	4.32	2.65	3.63	4.69	4.48

Table 6. Experimental and simulation results for the route repair topology.

		SHR-M			***SHR***	
	Experimental	***Simulation***	***Sim. with skip***	***Experimental***	***Simulation***	***Sim. with skip***
Packets sent	468	674	241	3,368	1,723	1,748
Packets received	1,232	1,350	607	8,563	3,411	3,643
Packet ratio (receive/send)	2.63	2.00	2.52	2.54	1.98	2.08
Delivery rate	24%	54.5%	2.67%	78%	66.1%	48.8%
End to end delay	48.49 ms	58.4 ms	25.9 ms	128.16 ms	129.6 ms	148.1 ms
Average hop count	5	5.80	4.74	6.10	5.96	5.72

4.1 Large scale simulations

The second set of simulations was performed using large scale networks and to compare the SHR, SHR-M and AODV protocols. The base configuration consists of a 2000 x 2000 ft^2 terrain populated with 500 motes, each of which has a nominal transmission range of 250 feet. The wireless medium was simulated with the free space propagation model [22].

The simulated WSN application sends packets with a mean size of 1000 bytes at a mean interval of 40 s. Each simulation was executed ten times with different random number seeds.

We performed five 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 mote; (ii) the rate of permanent mote failures; (iii) the rate of transient mote failures; (iv) the number of source-destination pairs; (v) the number of sources communicating with a single destination. SHR used a maximum *IgnoreCount* of 9 and a maximum hop equal to the distance to the destination plus $\log_2$ of the distance to the destination. A larger ratio would cause SHR to not drop packets that are not productive resulting in more transmitted packets per hop. SHR-M and SHR used a λ of 100 ms.

One of the statistics that we gathered was the number of packets transmitted per useful hop. A hop is considered useful only if contributes towards the first delivery of a packet at the destination. Duplicate packets that arrive at the destination do not contribute to the number of useful hops.

Effect of mote density. In this test, we varied the network density so that the mean number of neighbors per mote varied from 10 to 20 motes. The availability of more neighbors generally increases the physical distance a packet can travel at each hop. However, the probability of collisions also increases as more neighbors potentially compete to forward a packet. SHR-M and SHR have virtually the same hop count, so they are probably using the same route, which makes sense because the topology is static. The ratio of maximum hop to expected hop was set too high for the dense topologies that were simulated. A smaller ratio would cause SHR to more aggressively drop packets that are not productive resulting in fewer transmitted packets per hop.

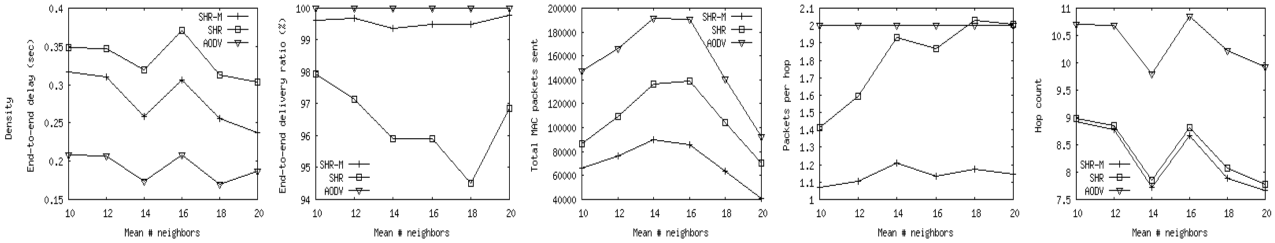

Fig. 5. Protocols' performance based on the mean number of neighbors per mote (network density).

Effect of mote failure rate. We tested two mote failure models, *transient* and *permanent*. There are several possible causes for transient mote 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 mote 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.

AODV is impacted strongly by topology changes. Link layer failure causes AODV to flood looking for a new route. The flooding may stop after a couple of motes, but it's still disruptive. However, the peaks are caused by outliers in one of the seeds used to generate the results.

SHR-M does not repair routes with any failures. Yet, repeating the DREQ/DREP phase periodically will improve the delivery rate and still use fewer packets than AODV.

SHR is somewhat affected by transient failures (95% delivery rate drops to 80%) but transmits much fewer packets than AODV. As the transient failure rate increases, it will overcome SHR's ability to repair routes.

Effect of traffic volume. We performed two tests to vary the volume of network traffic. The first test increased pairs of motes that communicated with each other exclusively. The second test increased the number of source motes communicating with a single destination mote, a situation that is common in wireless sensor networks. The results of these tests are shown in Figures 8 and 9, respectively.

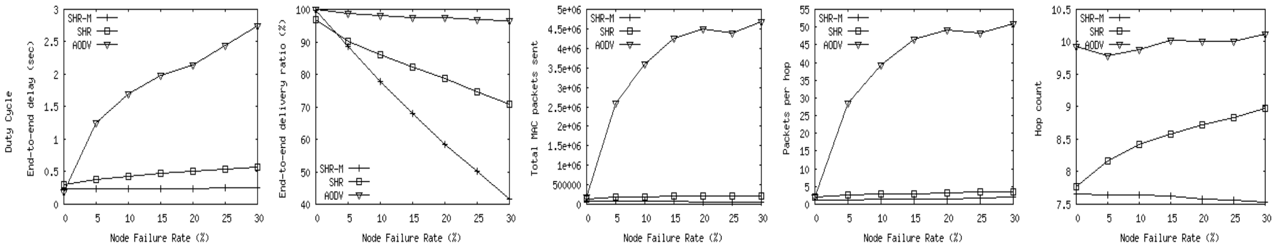

Fig. 6. Protocols' performance based on the motes' transient failure rates.

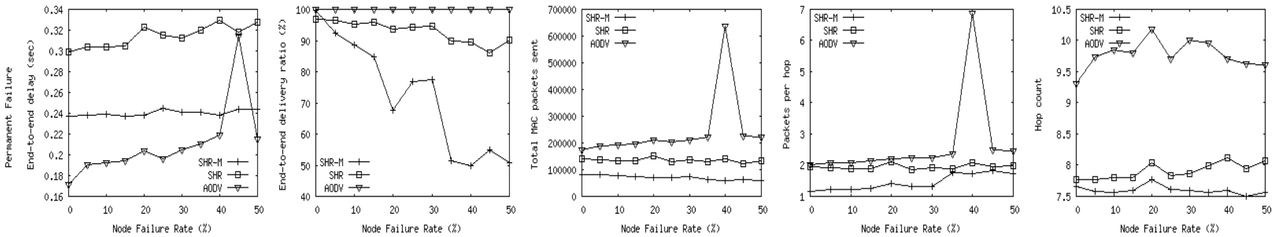

Fig. 7. Protocols' performance based on the motes' permanent failure rate.

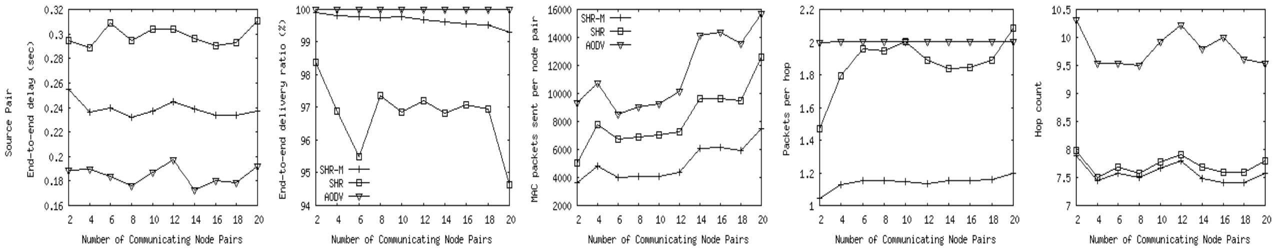

Fig. 8. Protocols' performance based on the number of source-destination pairs.

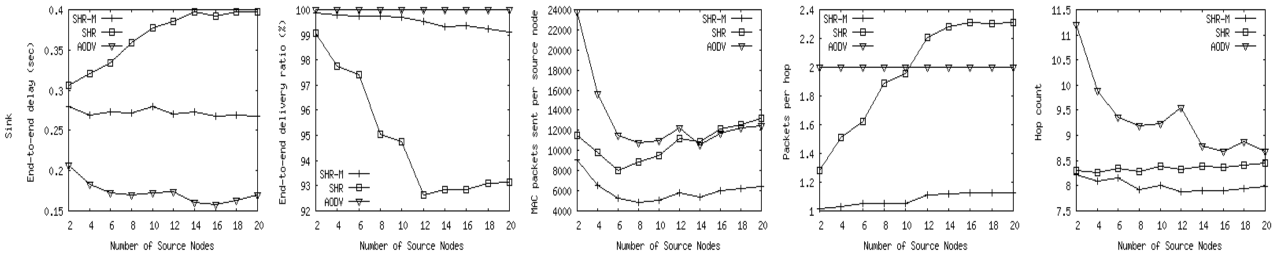

Fig. 9. Protocols' performance based on the number of sources communicating with one destination.

5. CONCLUSIONS

This paper describes our attempt of using a real implementation of a WSN routing protocol to verify its simulations, which then can be used to understand the protocol behavior in large scale networks. After analyzing the results of our implementation, reported in Section 3, we have extended the SENSE simulator to match the simulations with those results. We extended the sensor network model represented by the simulations with (i) a realistic link channel model which provides 90% connectivity to the direct neighbors, and (ii) a super hop possibility, in which a packet can be delivered to the distant (indirect) neighbor with probability of 5%. Thanks to these extensions, as reported in Section 4, we obtained good match of results of implementation and simulations on the same scenarios. Then, we used the extended simulator to study the protocols on large networks that would be difficult and costly to assemble and run. The conclusions from these simulations, presented in Section 4, follow.

The theoretically elegant and simple SHR-M protocol yielded poor results in real life test unless there was a high density of motes in the network. SHR performed well, but we learned that its λ must increase with mote density to counterweight increasing probability of collision of mote responses. MintRoute proved to be by far too promiscuous in transmitting packets to be efficient. Extended simulations also demonstrated good performance of SHR versus AODV.

Acknowledgement. Research was sponsored by US Army Research laboratory and the UK Ministry of Defence 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 U.S. 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. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, 40(8), 102-114 (2002).
2. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proc. ACM SenSys '03*, 14-27 (2003).
3. J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. ACM SenSys '03*, 1-13 (2003).
4. G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proc. 7th ACM Int. Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 174-181 (2004).
5. C. Perkins, E. Belding-Royer, and S. Das. RFC 3561-ad hoc on-demand distance vector (AODV) routing [Online], http://www.faqs.org/rfcs/rfc3561.html.
6. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proc. ACM MobiCom*, 56-67 (2000).
7. Y. Xu and W.-C. Lee, "Exploring spatial correlation for link quality estimation in wireless sensor networks," in *Proc. IEEE PERCOM '06*, 200-211 (2006).
8. The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/index.html.
9. G. Chen, J. Branch, M. Pflug, L. Zhu, and B. Szymanski, "SENSE: a wireless sensor network simulator," in *Advances in Pervasive Computing and Networking*, New York: Springer, 249-267 (2005).
10. Crossbow Technology, Inc., http://www.xbow.com.
11. TinyOS Community Forum, http://www.tinyos.net.
12. R. Poor, "Gradient routing in ad hoc networks," unpublished.
13. F. Ye, G. Zhong, S. Lu, and L. Zhang, "Gradient broadcast: a robust data delivery protocol for large scale sensor networks," *ACM Wireless Networks*, 11(2) (2005).
14. M. Heissenbttel, T. Braun, T. Bernoulli, and M. Waelchli, "BLR: beaconless routing algorithm for mobile ad hoc networks," *Elsevier's Computer Communications Journal*, 27(11) (2004).
15. M. Zori and R. R. Rao, "Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, 2(4), 337-348 (2003).
16. B. M. Blum, T. He, S. Son, and J. A. Stankovic, "IGF: a robust state-free communication protocol for sensor networks," *Technical Report CS-2003-11*, University of Virginia (2003).
17. Y. Xu, W.-C. Lee, J. Xu, and G Mitchell, "PSGR: priority-based stateless geo-routing in wireless sensor networks," in *Proc. IEEE Conf. Mobile Ad-hoc and Sensor Systems*, (2005).
18. D. Chen, J. Deng, and P. K. Varshney, "A state-free data delivery protocol for multihop wireless sensor networks," in *Proc. IEEE Wireless Communications and Networking Conference*, (2005).
19. M. Lisee, "The SHR Family of Protocols," available at http://www.ita.cs.rpi.edu, (2007).
20. M. Maróti, B. Kusy, G. Simon and A. Lédeczi, "The Flooding Time Synchronization Protocol," In *Proc. 2nd Int. Conf. Embedded Networked Sensor*, Baltimore, MD, USA, 39 – 49 (2004).
21. Elson, J. E., Girod, L., and Estrin, D. "Fine-Grained Network Time Synchronization using Reference Broadcasts," *Proc. 5th Symp. Operating Systems Design and Implementation (OSDI)*, 147–163 (2002).
22. T. S. Rappaport. Wireless Communications: Principles and Practice, Prentice Hall, 1992.