Biological Inspiration in the Improvement of Self Healing Routing

by

Christopher F Morrell

Summer 2007

Research Project

Approved:

Boleslaw Szymanski, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

August, 2007

CONTENTS

LIST OF FIGURES

ABSTRACT

Self Healing Routing (SHR) is a wireless network protocol developed to allow wireless sensor networks to route traffic based on minimal information about the network around them. In an effort to improve this protocol, several iterations of improvement have been developed based on real-world political and biological inspired models to improve the efficiency of the SHR protocol. The result of these changes is an overall improvement in the performance of the protocol, both in speed and reliability.

1. Introduction

As originally designed by Branch, Lisee, and Szymanski, Self-Healing Routing (SHR) [1,2] is a protocol customized for use in wireless sensor networks that allows sensors with minimal network awareness the ability to build and repair routes throughout their network. SHR is designed as a three phase protocol in which a node discovers a valid route to a destination (DREQ), the destination replies back to the source node (DREP), and data is transmitted. It is designed to use broadcast communication and a prioritized slotted transmission back-off delay scheme. This scheme allows nodes to use packet hop-count, and their distance to the destination to decide if they are on the best route within a flow, and allows them to act accordingly by either attempting to forward packets or not.

2. Political Incumbent Based Model

The initial inspiration for improvement to the SHR protocol came from a direct correlation to political elections. It is commonly known that for a variety of reasons, including name recognition and proven job performance, an incumbent candidate has an advantage over his competitors in an election. This incumbent advantage was not reflected in the original design of SHR. Previously, as nodes conducted their election process, they were not aware of any previous elections that they had won; therefore the self-imposed forwarding delay was randomly selected from the same range at each election. Using the model of a political incumbent, we allowed sensors to remember whether or not they had won the previous election in the form of a simple Boolean variable. If they had won the previous election, then they were allowed to decrease their delay by some fixed divisor, thus highly increasing the odds that they would win the following election.

At first glance, this seemed like a good idea, but upon further investigation and testing, it was found that cutting delays on the second election did not allow the routing to stabilize. This is because it sometimes takes several packets traversing the network before all nodes have completed their initialization and are prepared to pass regular traffic. Waiting for several packets to traverse also increases the probability that the chosen path is the best or near best path through the network. This model also did not provide any advantage to nodes that won multiple elections consecutively as the election memory was a Boolean, and only remembered if it had won the previous election.

3. Biological Based Model

Although the idea of the political incumbent model provided a start to improving the speed and reliability of the SHR protocol, it still left many issues that did not allow it to perform at its peak potential. In order to improve the idea more, inspiration was taken from ants and their use of pheromones to mark their paths. As ants find a path to a desirable destination, they mark their paths with pheromones, allowing other ants in the colony to utilize the same path. The more a path is used, the stronger the pheromones on that path are, and the more likely following ants will use that same path. As time passes, and a previously used path is bypassed, the pheromones fade, and following ants are less likely to use that path. A packet traverses a wireless network of sensors in much the same way as an ant traverses a series of paths on a forest floor. Both must choose from among many divergent paths that may or may not eventually get them to their desired destination. Also, if a good, uninterrupted path is found to the destination, it is beneficial to both the data packet and ant to follow that same established path.

In application of this idea, the variable that tracks the previous wins of each node was changed from Boolean to an Integer, starting with the value of zero for zero wins, and increasing incrementally with each consecutive win. The use of this integer very closely resembles the use of the ant's pheromones. This number was used as an exponent for the divisor of the delay, in order to shrink the delay increasingly after each consecutive election that a node won. The increasing divisor gave a marked advantage to nodes after several consecutive wins. Through extensive testing to decide the best divisor, it was found that raising 5 to the power of the

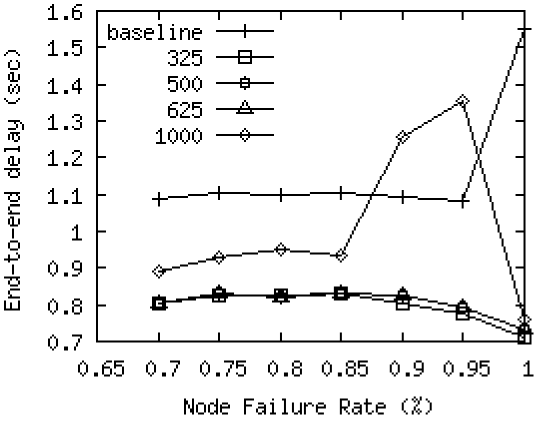

Figure 1: Node Failure Rate is in percentage of all nodes that did not fail. Failures are not permanent.

number of wins, and stopping at 625 gave the best result. This resulted in a series of divisors that consisted of 5, 25, 125 and 625. As shown in Figure 1, tests were conducted using 325, 500, 625, and 1000 as the final divisor in this series. As can be seen in this example of end to end delay, the baseline (no change to delay) requires approximately 1.1 second packet traversal time, while 325, 500, and 625 cut that time to between 0.7 and 0.8 seconds.

In addition to this, to overcome the problem of cutting delay on the first election that a node wins, it was decided that delay should not be cut until two elections are won. This amounts to the fourth packet across the network in the best case, as we have DREQ, DREP, first Data Packet (first election win), and second Data Packet (second election win). The extra win allows a flow to converge to a single path across the network before the delay is modified. The final result was a series of divisors that consisted of 25, 125, and 625. As long as a node maintained its ability to win elections, for each packet belonging to that flow that node's randomly selected delay was divided by 625.

4. Sequential Wins

Although the model described above resulted in significantly better speed and reliability, there is still no requirement that election wins come from consecutive packets within a flow. To correct this deficiency, an additional check was added before increasing the election win counter. This check ensured that the current packet was in the same flow and only one sequence number bigger than the last winning packet. The improvements in speed and efficiency using this improvement stems from the fact that occasionally nodes along two separate paths would declare themselves winners and packets would alternate between these paths. By ensuring that wins were from sequential packets, we

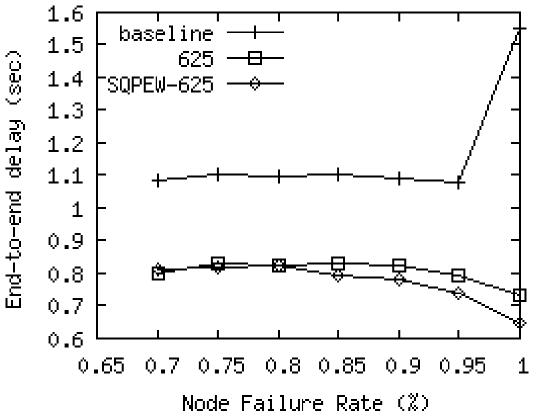

Figure 2: Node Failure Rate is in percentage of all nodes that did not fail. Failures are not permanent.

have forced the data to follow a single path. As shown in Figure 2, end-to-end transit time was decreased through the use of this method. Baseline shows the delay in the original design of SHR, 625 uses the biological model to decrease delay, and SQPEW-625 uses the same model, but ensures that wins are only counted for sequential packets within the same flow. Although the improvement in transit time is minimal when only 70 percent of the nodes are stable, in a 90 to 100 percent stable network, transit time improvement is approximately 0.10 seconds per packet.

5. Results

Using the Sense Sensor Network Simulator [3], final results show that by decreasing the delay on a continuously used path, we can both decrease transit time and increase reliability. Figure 3 shows graphs of all of the results using a simulation in which nodes fail temporarily. Figure 4 is the results of a simulation in which nodes fail permanently. Figure 5 involves a simulation in which there is a single sink node, and a range of 2 to 20 transmitting nodes. Figure 6 shows the results of the simulation in which there is a range of 2 to 20 transmitting pairs.

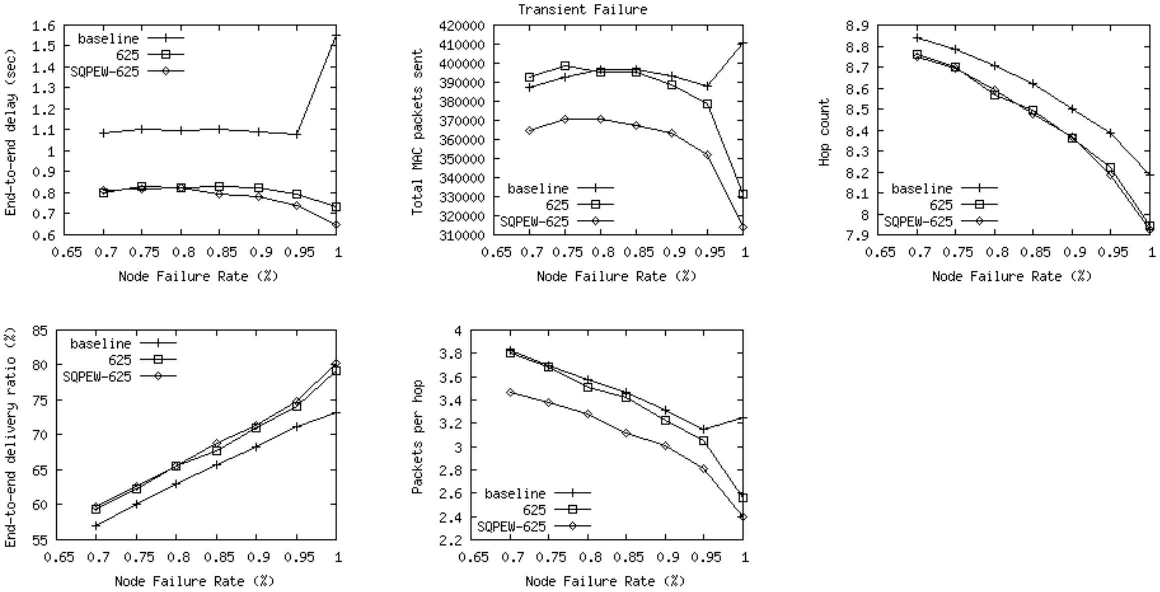

Figure 3: Node Failure Rate is in percentage of all nodes that did not fail. Failures are not permanent.

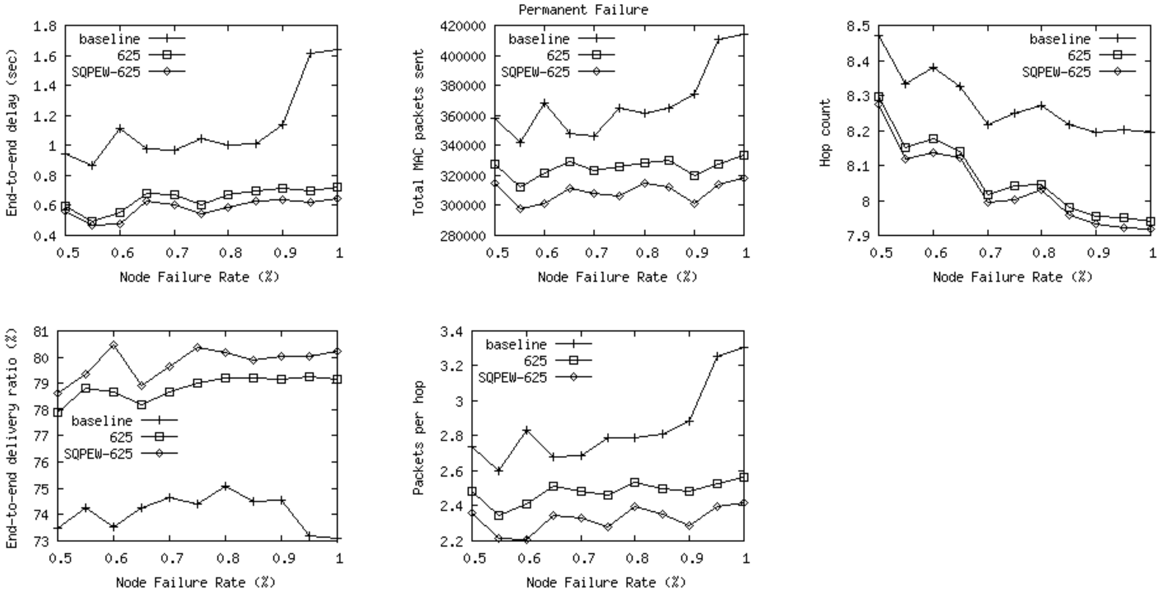

Figure 4: Node Failure Rate is in percentage of all nodes that did not fail. Failures are permanent.

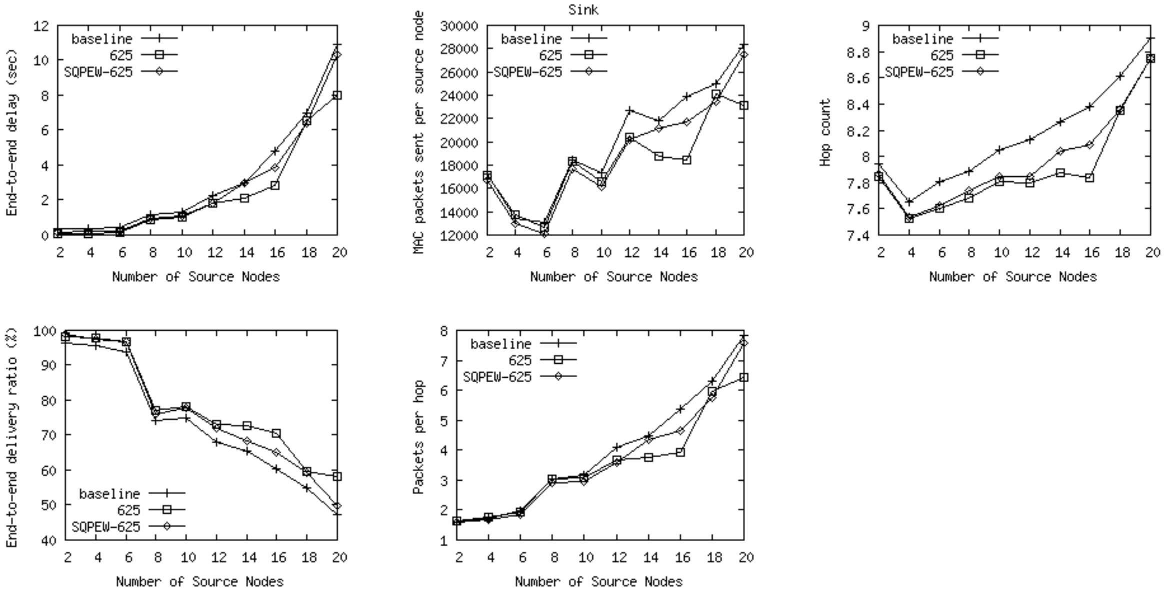

Figure 5: Single Sink Simulations

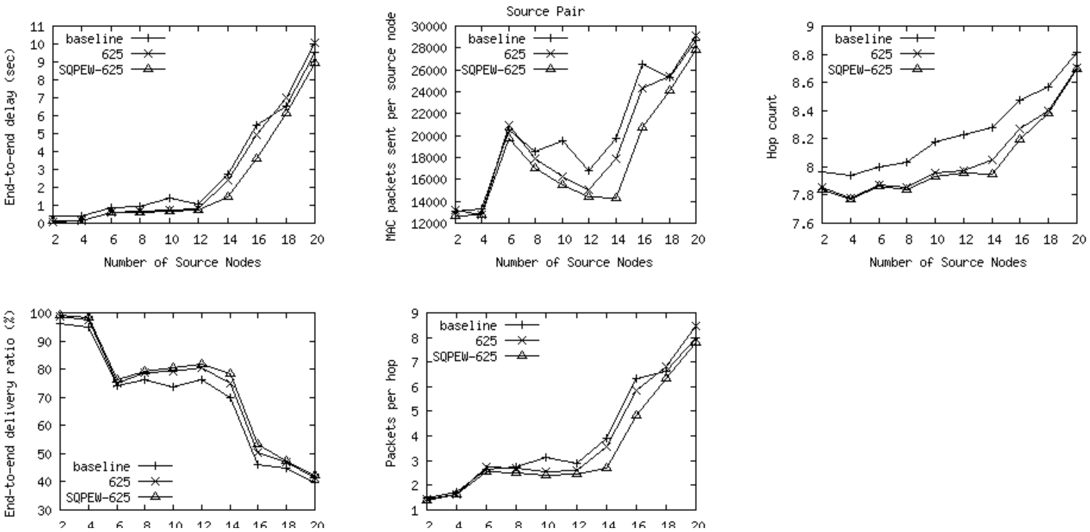

Figure 6: Varying pairs of communicating nodes

6. Problems

One of the largest downfalls of the improvements that we've made to SHR is the fact that we are trying to push more data through the network quickly. In the example of a single sink network with 500 nodes, we've found an average of 3-5 neighbors per node. This means that the 3-5 neighbors around the sink node must each pass traffic for approximately 100 nodes. In addition, each data packet must be acknowledged. This means that for each packet received, 1 acknowledge packet is sent, 1 data packet is sent, and 1 acknowledgement is received, thus quadrupling the volume of traffic across any given node, resulting in approximately 400 data packets handled within the 40 second average transmission interval, or 0.1 seconds per packet. Since we know that an even distribution is nearly impossible, this results in a high number of collisions, especially as we look closer to the sink node in a single sink environment. The same issues are observed in networks that have a higher number of source/sink pairs. On average, both sink and source-pair perform well up to 14 nodes, but beyond that, network degradation occurs exponentially. Both of these issues are evident in Figures 5 and 6.

7. Future Work

In an effort to improve the results from this work, the next steps will involve one of three ideas. The first of these is to combine packets. If a node that is preparing to send a packet receives a separate packet destined for the same node that it is currently sending to, it resets is timer, combines the packets, and prepares to send again. This will continue until the sending timer expires and the packet is actually forwarded. The second idea involves the idea of adding some small level of neighbor awareness to the nodes. This would give the nodes some visibility of what is going on around them, thus allowing nodes to work together to improve the efficiency of the network, rather than working completely independently as they currently do. The last of these three ideas involves the idea of shutting down nodes that sense large amounts of traffic around them, thus decreasing the amount of traffic in high density areas of the network rather than increasing the probability of collisions.

References

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

[2] A Self-selection Technique for Flooding and Routing in Wireless Ad-hoc Networks
G.G. Chen, J.W. Branch, and B.K. Szymanski
Journal of Network and Systems Management, vol.14, no.3, September 2006, pp. 359-380.

[3] SENSE: A Wireless Sensor Network Simulator
G. Chen, J. Branch, M. Pflug, L. Zhu, and B. Szymanski
Advances in Pervasive Computing and Networking, Springer, New York, NY, 2004, pp. 249-267.