Obtaining Robust Wireless Sensor Networks through Self-organization of Heterogeneous Connectivity

Abhinay Venuturumilli
Department of ECECS
University of Cincinnati
venutua@ececs.uc.edu
Ali Minai
Department of ECECS
University of Cincinnati
aminai@ececs.uc.edu

1. Introduction

A Wireless Sensor Network (WSN) is a set of sensor nodes that can communicate wirelessly with each other across an extended environment. Sensor networks are being investigated for various military, environmental, human-centric and robotic applications [Arampatzis 2005]. Most of the research on WSNs is focused on networks with identical nodes that have the same transmission range. This creates a homogeneous network whose connectivity can be modeled as an undirected graph. Homogenous networks are simple to analyze, but are well-known to be suboptimal with regard to efficiency, longevity and robustness [Yarvis 2005].

In a homogeneous network, the random deployment of nodes results in an uneven connectivity with critical nodes, making the network non-robust to node failure. A simple solution to overcome this problem would be to increase the transmission range of all nodes, but, this has the side-effect of creating undue congestion in other parts of

the network. In a heterogeneous network, in contrast, nodes can individually select their transmission range and can tune their connectivity locally without creating undue congestion. This effectively reduces the number of hops between nodes without increasing bandwidth needs and energy. Though the resulting networks are more efficient and robust than their homogeneous counterparts, they are difficult to analyze (see Duarte-Melo *et al.* [Duarte-Melo 2002] for some analysis).

Motivated by the considerations discussed above, several researchers have proposed algorithms for obtaining efficient heterogeneous networks. Ramanathan *et al.* [Ramanathan 2000] have developed the Local Information No Topology (LINT) and Local Information Link-state Topology (LILT) algorithms. Borbash *et al.* [Borbash 2002] discuss a distributed relative neighborhood graph algorithm (Dist_RNG) that minimizes congestion while maintaining connectivity. Liu *et al.* [Liu 2003] present a distributed algorithm for networks with nodes of different maximum transmission ranges. Ning *et al.* [Ning 2004] propose the Directed Relative Neighborhood Graph (DRNG) and Directed Local Minimum Spanning Tree (DLMST) algorithms that enable the network to reduce congestion compared to homogenous networks. However, these networks are not generally robust to random node failure. Some of the algorithms listed above have been compared and analyzed by Srivastava *et al.* [Srivastava 2003]. Recently, our laboratory has developed an algorithm for obtaining heterogeneous networks based on reverse engineering [Ranganathan et al, 2006].

In this paper, we propose a distributed algorithm to design a robust and energy efficient network. The nodes choose different transmission ranges such that the congestion and mean path length between the nodes are minimized. The simulation results reaffirm that the networks obtained by our heuristic outdo the homogeneous networks on the basis of all performance measures, while still being robust to random node failure.

2. Network Model

The sensor network is modeled as a graph whose vertices (nodes) represent the sensors and edges indicate direct communication between the nodes. The nodes are deployed in a uniform random distribution with density λ in a 2-D unit square area. In heterogeneous networks, the edges are directed because of the asymmetric connectivity. We assume that the environment is obstruction-free and that each sensor is aware of its geometric coordinates.

The network model used in this paper assumes that every sensor node can choose from two or three transmission ranges. The low power transmission radius is termed the *whisperer* radius (r_w), the high power transmission radius the *shouter* radius (r_s), while the medium power transmission radius – for the three level model – is called the *speaker* radius (r_t). The *whisperer* radius is chosen as $r_w = \alpha\, r_p$ where α is a constant factor < 1 and r_p is the percolation radius for the network [Stauffer 1994]. The shouter radius is chosen as $r_s = \beta\, r_p$ where β is a constant > 1, and the speaker

radius as $r_t = \gamma r_p$ where $\alpha < \gamma < \beta$. The nodes that are present within *whisperer* range of a node are listed in its adjacency list A_w. The nodes between *whisperer* and *speaker* radii are stored in the *inner-ring adjacency list* A_i, and the nodes between *speaker* and *shouter* radii are stored in the *outer-ring adjacency list* A_o. Thus, the set of nodes present between *whisperer* and *shouter* radius of the given node comprise the set $A_r = A_o \cup A_i$, where A_r is termed the node's *ring adjacency list*. The current adjacency list A_c for a node depends on whether the node is a *whisperer*, *shouter* or *speaker*.

3. Basic Radius Adaptation Algorithm

The fundamental principle underlying the radius adaptation algorithm is for each node to choose the smallest possible radius while maintaining the connectivity achievable by choosing the maximal radius. The basic 2-level case is as follows:

3.1 Initial Setup

Once the nodes are deployed, each sensor node follows a sequence of instructions to gather information about its neighboring nodes. First, every node boosts its transmission range to r_s and broadcasts its coordinate values along with its randomly generated unique identification number, and then cuts-back to r_w. Upon receiving coordinate value information from each of its neighbors in *shouter* range, the node calculates the Euclidean distance from each of these nodes to itself. If the distance to a node is less than r_w, it adds that node to its *whisperer adjacency* list A_w and, if the distance is more than r_w, it appends that node to its *ring adjacency* list A_r. Apart from A_w and A_r, the node stores another list A_c, which keeps track of its current adjacent nodes depending on the transmission range it is currently using. Since all nodes initially are at *whisperer* range, A_c equals A_w. The initialization process concludes with the transmission of the current adjacency list A_c by each node to all its whispering neighbors A_w. By the end of the setup process, each node knows which nodes it can transmit to at *whisperer* and *shouter* ranges and also the current adjacent nodes of its *whispering* neighbors.

3.2 Basic Radius Update Procedure

Following the completion of the setup process, each node traverses through its list of ring adjacent nodes and checks if they are reachable through one of its *whispering* neighbors. If all the ring nodes are reachable through its whispering neighbors, becoming a shouter would not result in any additional connectivity for this node, and it remains a whisperer.

Figure. 1., shows the ring connectivity of a node along with its *whispering* neighbors in a 2-level model. Assuming node *a* to be the first node to run the algorithm before nodes *b, c* and *d*, it traverses through each of the nodes in its A_r and checks if all those nodes are present among the A_c nodes of *b*, *c* or *d*. If this is not so, it boosts its

transmission range to *shouter*. Once node *a* changes its transmission range, it first updates its current adjacency list A_c, and then sends the updated A_c to nodes *b*, *c* and *d*. Thus, if node *b* later runs the radius update algorithm, it will find that all nodes from *a* to *k* can be reached through the adjacency list of node *a*, and if it has no other ring-adjacent nodes or if all its other ring-adjacent nodes can be reached through its other adjacent node *h*, it can remain a *whisperer*. Once this algorithm is run by each node independently and asynchronously, each node is guaranteed to be connected to all its *shouter* range neighbors either directly or through a *whisperer* range neighbor.

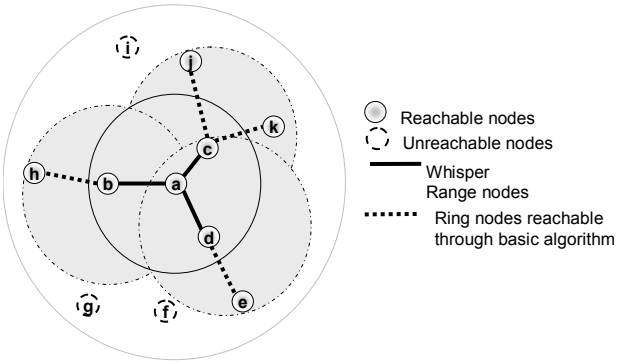

Figure. 1. Node at whisperer radius running basic update algorithm using knowledge about its adjacent node lists

3.3 Enhanced Radius Update Procedure

In the above procedure, each node checks only the adjacency lists of its immediate neighbors and if it finds any ring-adjacent node to be unreachable, it increases its transmission range to r_s without any further search. However, this can lead to an unnecessary increase of transmission radius since a node can be connected to its ring adjacent nodes through an alternative path of more than 2 hops. Therefore, an enhanced version of the algorithm implements a broader search by each node to find paths to all unreachable ring nodes through its reachable nodes. In Figure 1., node *a* is aware that all other nodes also have at least transmission range r_w. Thus, if it finds two nodes in its ring adjacency list which are within Euclidean distance r_w of each other, it knows that if it can reach at least one of the two nodes, it can also reach the other one through the first. At first, node *a* implements the basic update algorithm to determine which ring nodes it can reach through its adjacent nodes. It then checks if the remaining nodes in the ring are reachable through one of the reachable ring nodes by calculating the distance between ring nodes. If any new node is added to the list of reachable nodes after the verification, the remaining unreachable nodes' distance to this new reachable node is determined. This process continues until there is either no change in the reachable node list or all the ring nodes are reachable. If all the ring nodes are reachable, then the node decides to be a *whisperer*, but if it has one or more unreachable nodes in the ring, it decides to boost its transmission range to become a *shouter*.

3.4 Three Level Algorithm

In the algorithm described above, a node decides either to keep a *whisperer* transmission range or to change to a *shouter* transmission range depending on ring connectivity. An improved solution aims to achieve better congestion control by extending the transmission range to three levels viz., *whisperer*, *speaker* and *shouter*. In this algorithm, each node initially runs the original 2-level algorithm for the *speaker* and *shouter* transmission ranges. The node, thus decides either to remain at *speaker* range or to change to *shouter* range. If the node changes its transmission range to *shouter*, the decision is final, but if the node remains at *speaker* range, it goes through another decision cycle to decide if it can reduce its transmission range further to *whisperer* range. In the second cycle, it again runs the 2-level algorithm, but this time for the *whisperer* and *speaker* radii. If the node can ensure that it can reach its inner-ring adjacent nodes by transmitting at *whisperer* range, it reduces its transmission range to r_w. Thus, after the algorithm is run, there are three different transmission ranges in the network. The networks obtained by the 2-level and 3-level algorithms are compared against each other and with equivalent homogeneous networks. The results are discussed in the next section.

4. Results and Discussion

In the simulations, nodes are deployed in a unit square area with a uniform random distribution. The percolation radius is pre-evaluated for networks of different densities to determine the homogeneous and heterogeneous radius parameters. For the homogeneous case, all nodes are assigned percolation radius as the transmission radius. For heterogeneous networks, the coefficients α and β are taken as 0.8 and 1.25 respectively, and γ is assigned a value of 1, which means that the *speaker* transmission radius is equal to the percolation radius. For heterogeneous networks, the nodes are allowed to self-organize with the radius update algorithm. Once the algorithm is run by the nodes in the network, the self-organized heterogeneous networks (both 2-level and 3-level) and homogeneous networks are compared to check the following performance measures: 1) Maximum size of strongly connected component (SCC); 2) Congestion (mean in-degree) of the network; 3) Average inverse shortest path length (AISPL) among all node pairs; and 4) Mean transmission radius. Also, these homogeneous and heterogeneous networks are subjected to different levels of random node failure, and the effect on the maximum connected component size is compared to evaluate robustness.

The simulations were performed on networks ranging from 200 to 1000 nodes. For each of these network sizes, the performance measures were averaged over 100 different network configurations.

The comparison of strongly connected component sizes for intact networks is shown in Figure. 2., with the values normalized by the network size. Clearly, the size of the strongly connected component is consistently larger in heterogeneous networks when

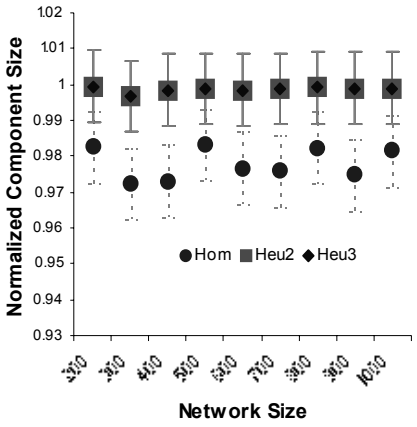

Figure. 2. Component Size with Error Bars

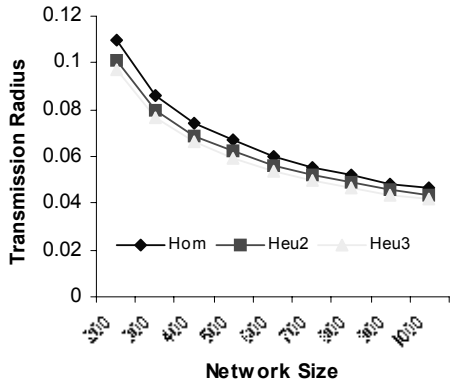

Figure. 3. Mean Transmission Radius

compared to homogeneous ones. By construction of the algorithm, the size of the SCC should be the same as what it would be if all the nodes in the network were *shouters*. Thus, it is observed that both 2- and 3-level algorithms have the same connected component size. The difference between the two algorithms can be seen in the other performance measures discussed next.

In Figure. 3., the mean transmission radius for the different cases is compared for networks of different sizes. It can be seen that, the mean radius is consistently lowest for the 3-level model and highest for the homogeneous case. Since the speakers in 3-level case would actually be shouters in the 2-level algorithm, the former leads to a lower mean radius than the latter. However, it is interesting to see that the mean transmission radii of both the heterogeneous algorithms are less than the percolation radius. At the same time, they also maintain very high connectivity compared to a homogeneous network. This illustrates the advantage of optimized heterogeneity over homogeneity.

The average congestion for networks of various sizes is shown in Figure. 4. It is clear that, like mean radius, congestion is lowest for the 3-level model; while the homogeneous case has the highest congestion. The congestion in a homogeneous network is high because homogeneity forces nodes in dense areas to have needlessly large radii and, therefore, high degree. The reason for higher congestion in the 2-level model compared to the 3-level model is that the nodes which choose to become speakers in the latter case have to become shouters in the 2-level model.

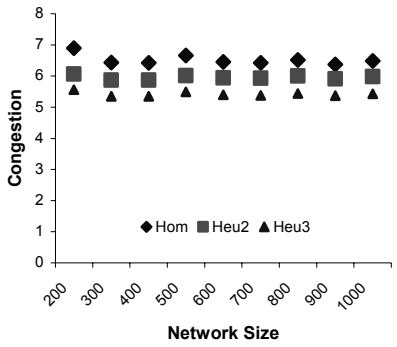

Figure. 4. Mean Congestion

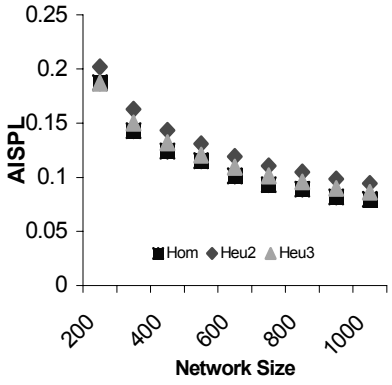

Figure. 5. Average Inverse Short Path length

One concern in optimizing transmission radii to minimize congestion without sacrificing connectivity is that it might lead to longer paths between nodes. To check this, we evaluated the networks for the average inverse short path length between all node pairs. We used inverse rather than the direct length to conveniently account for disconnected node pairs [Beygelzimer 2005]. Interestingly, though the heuristic networks have low mean radius and have less congestion compared to the homogeneous networks, they still have high AISPL. From Figure. 5., it can be seen that, the AISPL value is lowest (i.e., worst) for homogeneous networks while is highest for the 2-level model. The reason for the high AISPL in the 2-level model is the greater fraction of shouters, some of which become speakers in the 3-level case.

The percentage of shouters in the networks is shown in Figure. 6. It can be observed that, the fraction of shouters in the 2- and 3-level models is approximately 25% and 10%, respectively, and is independent of the network size; while the speaker fraction in the 3-level model is approximately 15%. Thus, the sum of the shouter and speaker fractions in the 3-level model is the same as the shouter fraction in the 2-level model.

Another important concern in highly optimized heterogeneous systems is robustness to random node failure [Dekker 2004, Crucitti 2004, Paul 2004], since optimization tends to "squeeze out" all the redundancy in the system. To evaluate robustness, we subjected both homogeneous and heterogeneous networks to random node failure. Node removal was sampled over 10 independent instances for each network layout. Robustness, measured by the size of the largest strongly connected component in the damaged network, was evaluated for 200, 300 and 400 node networks, though results are shown for 300 nodes only. From Figure 7, it can be observed that heterogeneous networks are more robust than homogenous ones even for 35% node failure. It can also be observed that robustness for 3-level networks drops at a faster rate than for

the 2-level case, which again reflects the presence of a larger *shouter* population in the latter.

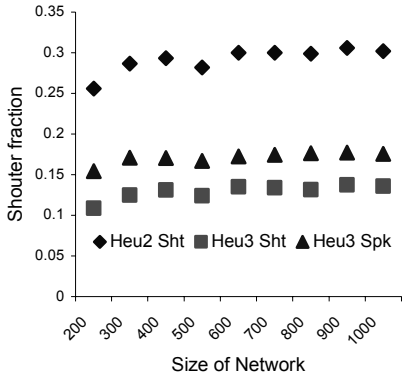

Figure. 6. Shouter Fraction

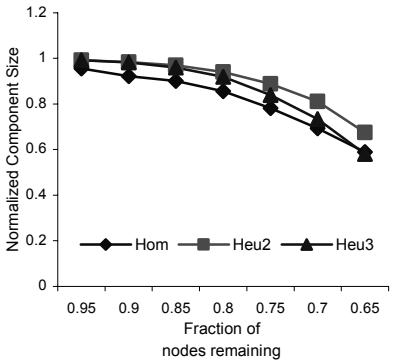

Figure. 7. Robustness for 300 size network

5. Conclusion

In this paper, a distributed heuristic algorithm and its improved versions were presented for the self organization of two and three radius heterogeneous sensor networks. In the proposed distributed scheme, each node independently runs the same heuristic and decides its transmission range depending on the topology of its local neighborhood. It has been shown that the networks obtained by this heuristic surpass homogenous networks on several performance metrics that were discussed above viz., Connectivity, Mean transmission radius, Congestion and AISPL. Also, it has been shown through simulations that the heuristic networks are more robust than the homogenous networks in the presence of random node failures. Thus, the proposed heuristics represent a very useful approach that enhances network efficiency while maintaining, and in fact greatly improving other critical performance metrics.

References

Arampatzis, Th., Lygeros, J., & Manesis, S., 2005, *A survey of applications of wireless sensors and wireless sensor networks*, Proceedings of the 13th Mediterranean Conference on Control and Automation (Limassol Cyprus)

Paul, G., Tanizawa, T., Havlin, S., & Stanley, H.E., 2004, *Optimization of robustness of complex networks*, The European Physical Journal B, **38**, pp. 187-191

Ning Li & Jennifer, C. H., 2004, *Topology control in heterogeneous wireless networks: problems and solutions,* Proceedings of IEEE INFOCOM 2004 (Hong Kong).

Yarvis, M., Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y., & Singh, S., 2005, *Exploiting heterogeneity in sensor networks*, INFOCOM, Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Miami).

Duarte-Melo, E., & Liu, M., 2002, *Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks* Proceedings of IEEE Globecom'02, (Taiwan).

Stauffer, D., & Aharony, A., 1994, *Introduction to Percolation Theory*. London: Taylor & Francis

Ramanathan, R., & Hain R, 2000, *Topology control in multihop wireless networks using transmit power adjustment*, Proceedings of IEEE INFOCOM 2000 (Israel), pp 404-413

Ranganathan, P., Ranganathan, A, Berman, K. & Minai, A.A., 2006, *Discovering adaptive heuristics for ad-hoc sensor networks by mining evolved optimal configurations*, Proceedings of the 2006 World Congress on Computational Intelligence (Vancouver, Canada)

Borbash, S.A., & Jennings, E., 2002, *Distributed topology control algorithm for multihop wireless networks*, Proceedings of the 2002 World Congress on Computational Intelligence (Honolulu, HI).

Srivastava, G., Boustead, P., & Chicharo, J.F., 2003, *A comparison of topology control algorithms for ad-hoc networks*, Proceedings of the 2003 Australian Telecommunications, Networks and Applications Conference, (Melbourne).

Liu, J., & Li, B., 2003, *Distributed topology control in wireless sensor networks with asymmetric links*, Proceedings of IEEE Globecom'03, Wireless Communications Symposium (San Francisco), pp 1257 – 1262.

Beygelzimer, A, Grinstein, G.E., Linsker, R, Rish, I, 2005, *Improving network robustness by edge modification*, Physica A **357**, pp 593 – 612

Dekker, A.H. & Colbert, B.D., 2004, *Network robustness and graph topology*, Proceedings of the 27th Australasian Computer Science Conference, (New Zealand), Estivill-Castro, V., ed., Conferences in Research and Practice in Information Technology, **26**, pp 359–368.

Crucitti, P., Latora, V., Marchiori, M., & Rapisarda, A., 2004, *Error and attack tolerance of complex networks*, Physica A **340**, pp 388 - 394