Self-Organization of Connectivity and Geographical Routing in Large-Scale Sensor Networks

Vinod Subramanian, Rajkumar Arumugam, and Ali A. Minai
Complex Adaptive Systems Laboratory
ECECS Department
University of Cincinnati
Cincinnati, OH 45221
ali.minai@uc.edu

A large-scale sensor network (LSSN) is formed when a very large number of sensor nodes with short-range communication capabilities are deployed randomly over an extended region. The random distribution of nodes in an LSSN leads to regions of varying density, which means that if all nodes have an identical transmission radius, the effective connectivity would vary over the system. This leads to inefficiency in energy usage (in regions of unnecessarily high connectivity) and the danger of partitioning (in regions of low node density). In this paper, we propose a technique for adapting a node's transmission radius based on a node's local information. Through localized coordination and self-organization, nodes try to attain fairly uniform connectivity in the system to aid in efficient data messaging in the system. We study the benefits of network adaptation by incorporating it into an adaptive geographical routing algorithm called *corridor routing*. We present simulation results showing significant improvement in performance over routing algorithms that do not use network adaptation. We also propose and study several scenarios for network adaptation in the presence of node failures, and explore the effect of parameter variation.

Introduction

A large-scale sensor network (LSSN) is formed when a very large number of sensor nodes with short-range communication capabilities are deployed randomly over an extended region. Unlike custom-designed networks, these randomly deployed sensor networks need no pre-design, need very little or no human supervision and configure themselves through a process of self-organization. These nodes are usually battery-operated and have limited computational capabilities. The vision motivating LSSN's is that using a large number of randomly deployed, locally communicating, cheap, disposable, simple and, therefore, individually unreliable nodes can produce more robust, flexible and scalable performance than a system of fewer, more powerful but expensive nodes without an appreciable loss of reliability. The focus, therefore, is in making the individual nodes as simple as possible while keeping the system "smart" [4, 6].

The problem of determining the optimal transmission radius for nodes in a wireless network is a key one for LSSNs. Researchers have argued that transmission power for each node must be sufficient to reach a "magic number", n^*, of other nodes in order to maximize forward progress of messages, minimize congestion, and minimize the possibility of partitioning. Kleinrock and Silvester [8] arrived at the "magic number" 6 as the optimal number of terminals to be covered by one transmission. In [13], Takagi and Kleinrock re-considered the problem and arrived at a new magic number nearly equal to 8. In LSSNs, a further reason to use no more transmission power than necessary is to conserve energy, since nodes have limited battery life.

In this paper, we propose a technique for adapting a node's transmission radius based on a node's local information. Through localized coordination and self-organization, nodes try to attain fairly uniform connectivity in the system. The proposed adaptation technique is a network level adaptation independent of the overlying routing algorithm. Our approach is based on the principle — derived from complex systems such as cellular automata and neural networks — that decisions made by nodes must be based on simple, *locally available* information rather than awareness of the wider network. In the approach proposed in this paper, we utilize the location information of nodes, which might be obtained using the global position system (GPS) or some other localization scheme [9, 3].

System Model and Problem Statement

The network we consider comprises *n* nodes distributed uniformly in a two-dimensional region. Each node, limited by its energy capacity, can only communicate over a limited *transmission radius*. It is, thus, connected only to a small subset of the nodes, called its *neighbors*. In this paper, we assume that the nodes have variable transmitter power and receiving sensitivity, so a node's neighbors can change. The wireless network is, therefore, a random graph $G = (N, E)$, where the nodes are $N = 1, .., n$ and there is an edge $(i, j) \in E$ if node i is a neighbor of node j in the wireless network. Initially, we assume that all nodes have identical transmission radius, the graph is undirected.

Messages in our system are sent to geographical locations (coordinates) rather than to specific nodes. We assume that every node knows its own position. Using an ini-

tial setup process that is repeated periodically, each node in the system pre-determines the coordinates for which it is the closest node. Thereafter, it considers any message directed at any of those coordinates as intended for itself. Henceforth, we refer to this node as the *destination node*, even though the source node does not explicitly address the message to it. All data messages have a header with source and destination coordinates.

Medium Access Control

The Medium Access Control (MAC) protocol we use, is a modification of the Slotted Carrier Sense Multiple Access (CSMA) protocol [12] and is called Carrier Sense Multiple Access with Mini-Back off (CSMA-mb) [1]. In CSMA-mb, the channel is divided into major time slots. Each major time slot is subdivided into a number of mini-slots. The size of each mini-slot is the maximum propagation delay among any neighboring nodes in the system. Nodes contend for channel access at the first mini-slot of every major time slot. The first m mini-slots are reserved for contention in the system. At the beginning of every major slot, each node wishing to transmit backs off randomly to a value in the range of [1, m]. This process is termed as the mini-backoff scheme. At the end of its mini-backoff, the node senses the carrier again. If the channel is busy, it backs off to the next major slot; otherwise, it transmits. CSMA-mb thus follows a cautious persistent back off procedure. CSMA-mb has been shown to produce better results than slotted CSMA [1].

Network Adaptation Technique

The random distribution of nodes in an LSSN leads to regions of varying density. The network adaptation technique discussed in this section intends to adapt a sensor node's transmission radius based on its 1-hop neighborhood in order to attain a pre-determined connectivity in the system, thereby making the network more efficient and robust. This algorithm is explained in detail below.

The Algorithm

The nodes start off with a uniform high radius of communication so as to ensure good connectivity in the system. There exists a setup period in the system, known as the *system setup time*, during which the network adaptation takes place. During this period, nodes broadcast only *hello messages*. These control messages are of a smaller packet size compared to data messages. The system set-up time can be divided into two stages:

1. **Neighborhood Identification Stage:** During this stage of the system set-up time, nodes identify their neighborhood configuration using the hello messages. Each hello message carries information about its node's geographical location and its current transmission radius. When a node A receives a hello message

from a new neighbor B, A responds by sending an *acknowledgement (ack)* message as reply. This type of control messaging is termed *event-driven control messaging*. In addition to this, the system also has *periodic control messaging*. The rate of this messaging is denoted by $\lambda_{control}$. With the help of both the event-driven and periodic control messaging, nodes develop and update lists of their 1-hop neighbors (locations). The entire process helps a node determine its local connectivity.

2. **Neighbor Pruning Stage:** The second stage of the system set-up time involves the actual adaptation process.Using the information obained in the identification stage, nodes adapt their connectivity to attain a pre-determined number of neighbors denoted by n^*. By making a simple computation, nodes either decide to increase their transmission power by a pre-determined value or decide to decrease their transmission power so as to reach n^* neighbors. Through a series of updated "hello" and "acknowledgement" messages, nodes self-organize their neighborhood and try to attain the n^* neighbors.

The neighborhood identification stage and the neighbor pruning stage together constitutes the system set-up time in our algorithm. We choose the pre-determined number of neighbors (n^*) in our simulations following the analysis in [8, 13]. We also propose and study several scenarios for network adaptation in the presence of node failures, and explore the effect of parameter variation.

Routing Algorithm

A primary requirement for any sensor network to function is the ability to transfer information between arbitrary points in the system, and in this paper, we study the benefits of network adaptation by incorporating it into an adaptive routing algorithm developed by us. This algorithm, called *corridor flooding*, is an intelligent broadcast algorithm that seeks to balance the redundancy and robustness of broadcast with the efficiency of directed routing. For LSSNs with a large number of very limited, unreliable nodes, we consider broadcast more appropriate than unicast because: 1) The computational resources needed for path discovery or creating and maintaining routing tables are not available to the nodes; 2) Most paths are too long (in terms of hops) to be discovered or maintained by a source node, and too transient to be worth discovering or maintaining. In this situation, a *broadcast-based* approach with its inherent redundancy of paths is the natural answer. However, it must be tempered by concerns of efficiency so that it is less wasteful than simple flooding.

Our approach is based on the principle that decisions made by nodes must be based on simple, *locally available* information rather than awareness of the wider network. Thus, each node in our network, upon receiving a message not intended for it makes a simple decision: Should it re-broadcast the message? This is simpler, for example, than determining which node to forward the message to. By intelligently controlling the basis for each local re-broadcast decision, we arrived at the *corridor flooding algorithm*, which scales effectively and is robust against node failure. The corridor routing algorithm utilizes the geographical location information of the nodes to route messages.

We define a corridor as an *imaginary* two dimensional region of a pre-determined width extending from the message source to the message destination. The pre-determined width of the corridor is termed the *corridor width*. As the imaginary corridor depends on the locations of the source and the destination of a message, the length and orientation of the corridor differs for different source-destination pairs. The corridor for a particular message is consistent at all nodes involved in routing the message because the nodes infer the same corridor based on the information received in the message itself. The source and destination location for every message is encoded in the message header.

When a node receives a message, it evaluates the position of the imaginary corridor and computes whether it lies within the region. If it lies within the corridor, the node re-broadcasts the message; otherwise, the message is discarded. This way the nodes *contain* the flood of a message to the message's imaginary corridor. The primary parameter in this algorithm is the corridor width, w. The corridor width determines the number of redundant paths to the destination, with the consequent robustness against node failure. A wider corridor increases the available redundancy of paths between source and destination, but increases congestion and wastes energy. A narrower corridor has the converse effect. Thus, determining an optimal corridor width is crucial to obtaining a robust and efficient system in the face of node failures.

The corridor routing algorithm exploits the *best available redundancy* in the system to achieve superior performance over broadcast-based flooding. In this paper, we show that the corridor routing algorithm outperforms even unicast-based routing protocols in networks with high rates of node failure.

Comparison Protocols

In order to evaluate the performance of the corridor routing algorithm with the network-level adaptation, we have implemented three other routing protocols in this work. These are described below:

Simple Flooding: The classic flooding algorithm is the baseline case for comparison in our performance study. In this protocol, a non-destination node re-broadcasts any message it receives exactly once.

Pseudo Unicast: In *pseudo-unicast*, each non-destination node, upon receiving a message, unicasts the message to its *most forward neighbor* — the neighbor that provides the greatest progress towards the destination [13]. Nodes in pseudo-unicast do not employ any channel-reservation scheme [7, 2], and thus collisions are prevalent. If the most forward neighbor happens to be the immediate source node (evaluated with the information carried by the message), then the transmitting node chooses the *next* most forward node / the least backward neighbor to unicast the message. In essence, pseudo-unicast can be seen as *worst-case unicast* — the least sophisticated kind of unicast. It can also be seen as broadcast with forwarding only by a single node. By comparing this protocol with the corridor routing algorithm, one can quantify the value of path redundancy for the performance of the system: It shows whether a system with a single point of failure is sustainable under highly unreliable conditions or if it is necessary to preserve a certain degree of redundancy to achieve robustness under failures.

Super Unicast: *Super-unicast* works like pseudo-unicast, but assumes idealized conditions for communication. Hence, a transmission from a node A to a neighbor B is always successful, provided B is in commission. The motivation for investigating super-unicast is to compare the corridor routing algorithm with the *best-case unicast*. Although pseudo-unicast eliminates some of the problems of broadcast, it still suffers from the classical *hidden* and *exposed* terminal problems. Researchers over the years have proposed many sophisticated unicast protocols to address these problems, e.g. MACA [7] or MACAW [2]. Instead of implementing any of these sophisticated protocols, we simply consider the case obtained when such protocols work perfectly, obtaining super-unicast.

We believe that by comparing our scheme with both pseudo-unicast and super-unicast, we cover the entire spectrum of unicast protocols.

Simulation Framework

In this section, we describe the simulation framework and systematically present the results of performance of the corridor flooding algorithm in comparison with the simple flooding algorithm and unicast-based algorithms.

Performance Metrics

The following performance metrics are used to evaluate the various protocols:

1. **Message Delivery** The ratio of the number of messages that reached the destination node to the number of messages generated in the system.
2. **Transmission Ratio** The number of messages transmitted in the system for every message generated. This measures the efficiency of energy usage.
3. **Message Wait Time** The average wait time of messages at every hop, expressed in terms of mini-slots.
4. **Hop Length** The average number of hops taken by messages that have reached the destination.

Simulation Model

We performed a series of simulations with networks of 100, 500 and 1000 nodes. The nodes had an initial uniform transmission radius of 10 units, 4 units and 3 units for the 100-node, 500-node and 1000-node cases, respectively. They were distributed uniformly on a 50×50 square geographic region. The pre-determined corridor widths were chosen as 40, 16 and 12 for the 100-node, 500-node and 1000-node cases, respectively.

Messages were generated according to a Poisson process with message generation rate λ. The λ values used in simulation ranged from 0.05 to 0.25 (per slot). Every message generated had a randomly generated source-destination coordinate pair. The simulation made sure that the source and the destination were not identical. The $\lambda_{control}$

was fixed at 2.5 (per slot). We implemented a discrete-event simulator that kept track of message collisions in the system. The width of the major slot in CSMA-mb was 70 mini-slots, with m = 30 mini-slots dedicated for contention.

Test Scenarios

The performance of the protocols was evaluated under two different scenarios:

1. *Infinite Energy Assumption:* Nodes in the system are assumed to have infinite energy for communication. The nodes are thus in operation throughout the run. We explore this scenario to analyze the best-case performance of the protocols.

2. *Random failures:* This scenario considers the possibility where certain nodes in the system fail. To model this, we divide the nodes in the system into *stable* nodes and *normal* nodes. Stable nodes are ideal, infallible nodes. These nodes are in commission throughout the lifetime of the system. In contrast, a normal node fails in each major slot with some probability P_f. This failure could be attributed to wear and tear or shortage of energy or even a timeout period. The node could become active and participate in routing in the next major slot. The concept of stable nodes was introduced to study the robustness of the system in relation to the degree of unreliability prevailing in the system.

Simulation Results

Based on preliminary simulations, we determined that a value of $n^* = 8$ was appropriate for our networks. This was consistent with the recommendation in [8]. The choice of n^* and the density of nodes then determined the default radius of transmission, and thus the energy usage of nodes.

Adaptive Network: Infinite Energy Condition

Figures 1 and 2 show the performance curves of the 500 and 1000-node adaptive network under infinite energy conditions. In both cases, the broadcast-based algorithms out-perform the unicast-based algorithms in terms of message delivery. Even in larger networks, simple flooding performs exceptionally well in terms of message delivery. This is attributable to the better and fairly uniform connectivity in the system. Corridor routing performs either as well as or better than simple flooding. The savings in energy is more substantiable as the network becomes larger, emphasizing scalability. Also, there is a slight improvement in message latency in corridor routing as compared to simple flooding.

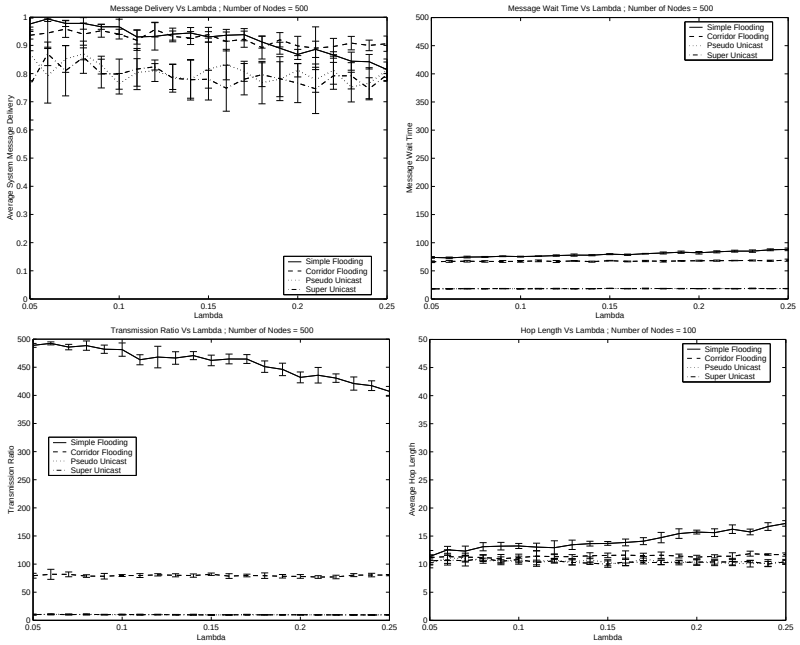

Figure 1: Adaptive network, infinite energy scenario: all protocols comparison with number of nodes = 500

Adaptive Network: Random Failures

The first part of the random failure study involved keeping the message generation rate (λ) constant and varying the percentage of stable nodes. Normal nodes in the system fail with a probability of 0.5 (i.e., $P_f = 0.5$) in every major time slot.

Figure 3 shows the performance curves for a 100-node network with the message generation rate fixed at $\lambda = 0.2$. Corridor routing shows the best message delivery and wait times not much worse than the unicast protocols. Energy usage is much better than simple flooding. Overall, corridor routing clearly provides the best combination of reliability, efficiency and robustness. It can be argued that using handshaking mechanisms similar to MACA [7] or MACAW [2], one could achieve higher message delivery even under the effect of node failures. However, these methods are not suitable for the simple nodes and the failure-prone scenarios being studied here.

The second study of the random failure mode involved fixing the percentage of stable nodes in the system while varying the message generation rate (λ). Again, the normal nodes fail with a probability of 0.5 in each major time slot (i.e., $P_f = 0.5$). The percentage of stable nodes is fixed at 50% for the simulations and λ is varied from 0.05 to 0.25.

Figure 4 shows the performance for a 500-node network. The adaptive unicast-based algorithms performs poorly in the face of such a high node failure rate (Only 50

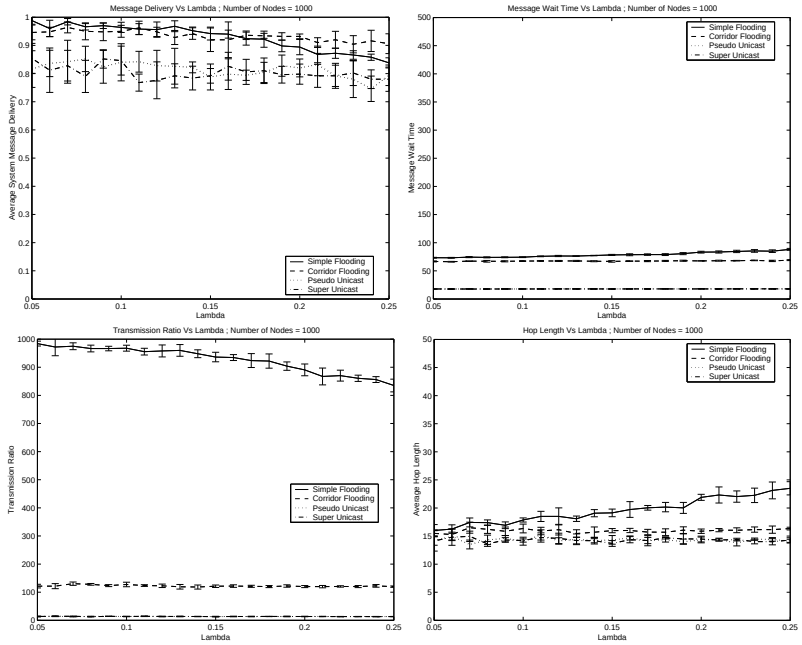

Figure 2: Adaptive network, infinite energy scenario: all protocols comparison with number of nodes = 1000

% of the nodes are stable). On the other hand, the broadcast-based protocols perform much better than their unicast counterparts under similar test conditions in terms of message delivery. While corridor routing performs almost as well as simple flooding in terms of message delivery, there is appreciable gain in the transmission ratio and a significant improvement in the message latency (product of message wait time and hop count). The unicast-based systems performs poorly because of the presence of a single point of failure of these algorithms coupled with the need for the messages to traverse a large number hops to reach their destination.

So far, in the analysis of the node failure modes, we used $n^* = 8$ as the pre-determined neighbor count. What happens if n^* were to be increased? A higher n^* would mean a greater radius of transmission for the nodes.

Next, we consider what happens if n^* is raised beyond 8. One might assume that there would be a corresponding increase in the amount of traffic received at each node. However, in the presence of node failures, such an increase in a node's neighborhood actually helps performance. We chose $n^* = 10$ as the new pre-determined neighbor count for this analysis. The value of n^* can be chosen based on the anticipated risk of failures of the nodes. In a scenario in which nodes have a high risk of failing, a slightly higher n^* than the magic number of 6 or 8 [8, 13] would aid in the messaging.

Figure 5 shows the performance comparison of the simple flooding and corridor flooding protocols with $n^* = 8$ and $n^* = 10$ cases under node failures. In this scenario,

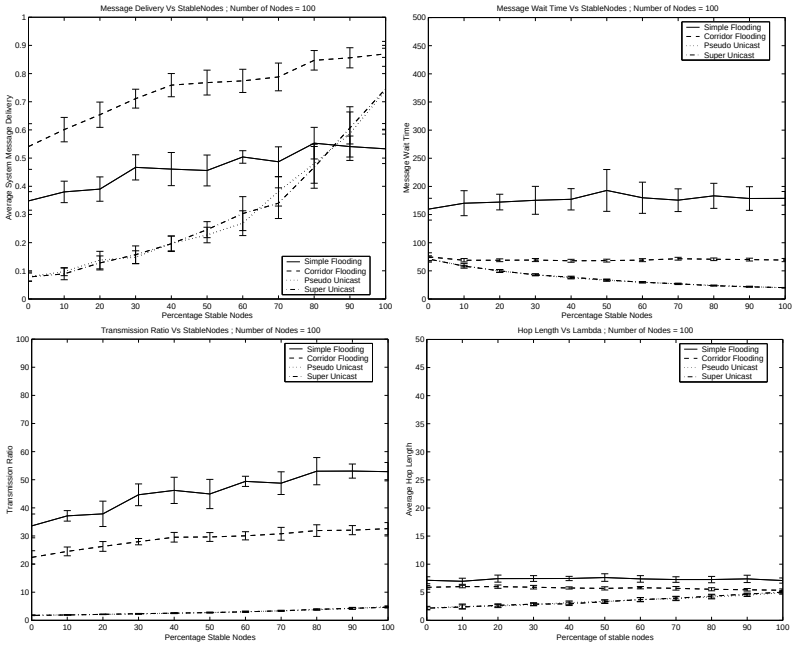

Figure 3: Adaptive network with node failures: all protocols comparison with number of nodes = 100 and $\lambda = 0.2$ (vs percentage of stable nodes. $P_f = 0.5$)

λ was fixed at 0.1, the percentage of stable nodes was varied from 0 % to 100 % and the probability of failure of the nodes $P_f = 0.5$. The message delivery of simple flooding with $n^* = 10$ shows up to 35 % improvement over the message delivery of simple flooding with $n^* = 8$. Also, there is a slight improvement in the performance of the corridor routing protocol in terms of message delivery. The width of the corridor is fixed as 40 in both cases; the transmission ratio of both the corridor routing curves almost overlap due to this reason. There is a slight increase in the wait times for the $n^* = 10$ case because of the increase in contention at each hop.

Essentially, the better results for $n^* = 10$ reflect an increased redundancy of paths that compensates for node failures. Of course, there is a trade-off here — increasing the n^* above a certain value will lead to performance degradation. We have not studied that phenomenon in this work.

Summary and Future Work

The work reported here shows that adaptive broadcast-based algorithms perform significantly better than unicast-based algorithms in networks of simple nodes with high failure rates. The primary drawback of the network adaptation is the need for the system set up time. The system set up time uses up both bandwidth and energy in the system.

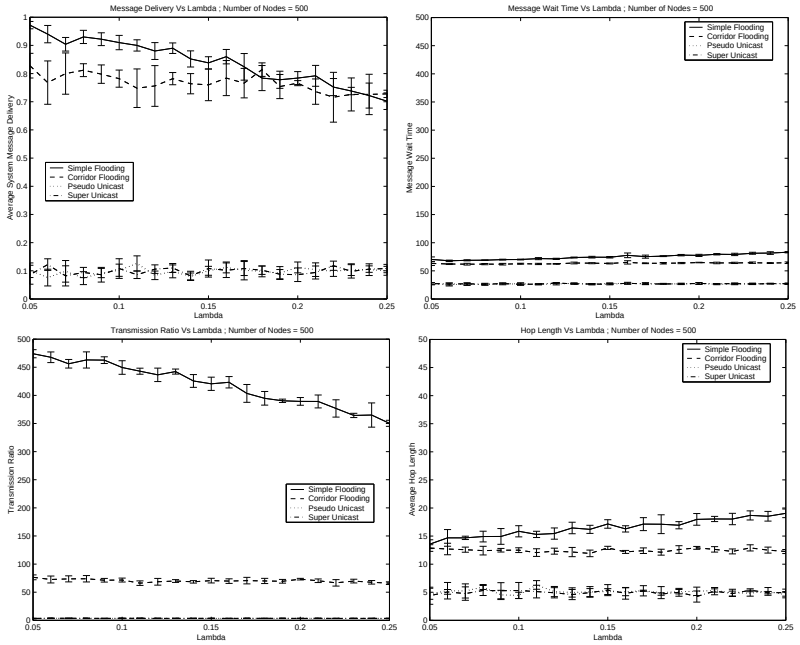

Figure 4: Adaptive network with node failures: all protocols comparison with number of nodes = 500, percentage of stable nodes = 50% and probability of failure $P_f = 0.5$

However, this would typically be only a small fraction of system lifetime.

Node mobility is an issue left un-explored. Actually, nodes in an LSSN are not expected to be highly mobile because of their limitations on energy, cost and complexity. Also, nodes in an embedded sensor network will generally be stationary. Nevertheless, as we rely on the geographical locations of the source and the destination nodes, node mobility would require that nodes continuously estimate their own coordinates in an absolute or relative system, e.g., using beacons or landmarks [9]. Also, the network adaptation technique would be drastically affected by node mobility because of the dynamics of node links.

Bibliography

[1] Arumugam R. , Subramanian V. and Minai A. A., "Intelligent Broadcast for Large-Scale Sensor Networks", *Proceedings of the 4th International Conference on Complex Systems* (this volume), June 2002.

[2] Bharghavan V., Demers A. J., Shenker S. and Zhang L., "MACAW: A Media Access Protocol for Wireless LAN's", *Proceedings of the ACM SIGCOMM*, 212–225, August 1994.

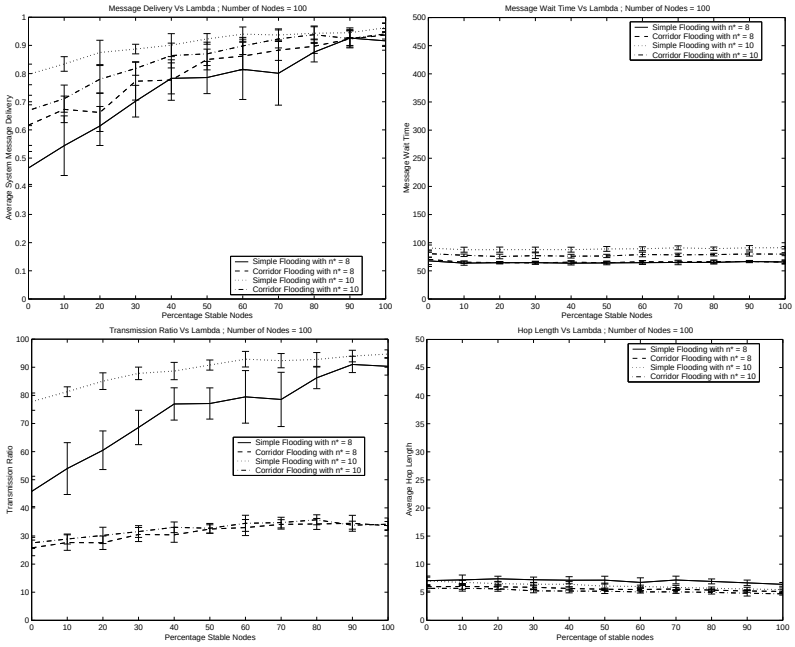

Figure 5: Adaptive network with node failures: comparison with higher n^* - simple flooding and corridor routing with $n^* = 8$ and $n^* = 10$ and $\lambda = 0.1$ (vs percentage of stable nodes. $P_f = 0.5$)

[3] Bulusu N., Estrin D., Girod L. and Heidemann J., "Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems", *Proceedings of the 6th IEEE International Symposium on Communication Theory and Application*, July 2001.

[4] Estrin D., Govindan R., Heidemann J. S. and Kumar S., "Next Century Challenges: Scalable Coordination in Sensor Networks", *Proceedings of the fifth annual ACM/IEEE International Conference on Mobile Computing and Networking*, 263–270, 1999.

[5] Johnson D. B. and Maltz D. A., "Dynamic Source Routing in Ad Hoc Wireless Networks", *Mobile Computing*. Kluwer Academic Publishers, 153–181, 1996.

[6] Kahn J. M., Katz R. H. and Pister K. S. J., "Next Century Challenges: Mobile Networking for Smart Dust", *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*", 271–278, 1999.

[7] Karn P., "MACA - A New Channel Access Method for Packet Radio", *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, September 1990.

[8] Kleinrock L. and Silvester J., "Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number", *IEEE National Telecommunications Conference*, 4.3.1–4.3.5, December 1978.

[9] Leonard J. L., "Large-Scale Concurrent Mapping and Localization", *Proceedings of SPIE Sensor Fusion and Decentralized Systems III*, 4196, 370–376, 2001

[10] Perkins C. E. and Bhagwat P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Computer Communication Review*, 24(4): 234–244, October 1994.

[11] Perkins C. E. and Royer E. M., "Ad-hoc On-Demand Distance Vector Routing", *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications* , 90–100, February 1999.

[12] Rom R., and Sidi M., "Multiple Access Protocols Performance and Analysis", *Springer-Verlag*, 1990.

[13] Takagi H. and Kleinrock L., "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals", *IEEE Transactions on Communications*, 32(3), 246–257, March 1984.