SCRIBE: SELF-ORGANIZED CONTENTION AND ROUTING IN INTELLIGENT BROADCAST ENVIRONMENTS

Rajkumar Arumugam, Vinod Subramanian and Ali A. Minai
Department of Electrical & Computer Engineering and Computer Science
University of Cincinnati
Cincinnati, OH 45221-0030

ABSTRACT

Large-scale sensor networks (LSSNs) are systems with a very large number of networkable sensors deployed randomly over an extended environment, rendering it observable. These randomly deployed networks need no pre-design and configure themselves through a process of self-organization. The purpose of our research is to consider how reliable, robust and scalable location-addressed communication can be assured in LSSN systems built from large numbers of inexpensive and unreliable nodes with limited capabilities using only broadcast communication. We present viable broadcast-based protocols for channel access and network organization. We focus on the situation where the transmission radius of individual nodes is much smaller than the size of the system, so that most messages need a large number of "hops" to reach their destination. In view of this, and given the unreliable nature of the nodes, we consider whether an inherently redundant intelligent broadcast scheme can provide sufficient message reliability. This bottom-up approach to communication is shown to provide reliable communication at the system level and to outperform more complex model-based approaches.

INTRODUCTION

With recent technological advances, it is possible to create ad-hoc smart environments using large numbers of very simple, low-energy sensor elements with rudimentary computation and communication capabilities [3, 4, 5, 6, 10, 15]. The sensors could provide active real-time on-demand information to a mobile user(s) for purposes such as path planning, navigation, assessment, surveillance, entertainment, etc., or to a user querying from a remote location. A canonical scenario is where sensors are dropped onto hostile terrain before sending in autonomous vehicles to search for targets or to navigate safely through the terrain. Self-organized sensor networks differ fundamentally from conventional wireless ad-hoc networks in the capabilities of their individual nodes, reliability, connectivity, coverage and robustness.

The purpose of our research is to explore how a system comprising a very large number of randomly distributed nodes can organize itself to communicate information between designated geographical locations. To keep the system realistic, we assume that nodes have only limited reliability, energy resources, wireless communication capabilities, and computational capacity. Thus, direct long-range communication between nodes is not possible, and most messaging involves a large number of "hops" between neighboring nodes.

Most of the work on ad-hoc wireless networks has focused on networks where message paths are only a few hops long. Data messages in such a system are typically unicast, i.e., they are between specific pairs of nodes. From a complex systems viewpoint, unicast-based methods do not sufficiently exploit the inherent parallelism of the system to achieve robustness — a critical issue in networks of unreliable nodes. Rather than using directed unicast between nodes, we study the possibilities of broadcast. A major advantage of broadcast is the lack of a complex network layer protocol for routing, address and location management. This is consistent with the idea of a simple system able to achieve global functionality through self-organization. Broadcast is also simple and inexpensive, and can exploit the inherent redundancy of paths in the network to achieve robustness. In some cases, broadcast is the only possible means of communication e.g., when sensors embedded in a conducting structure communicate through the structure.

In the simplest case, broadcast corresponds to flooding, where every message received by a non-destination node is "flooded" to all the node's neighbors. However, broadcast by flooding is extremely wasteful of resources and results in a lot of collisions — the so-called *broadcast storm problem* [11]. To overcome the problems of flooding while retaining its inherent parallelism, we explore methods for *intelligent broadcast*. In this approach, each node receiving a message decides whether to re-broadcast it to all its neighbors or to ignore it. Thus, efficiency in the messaging process is created by a node's decision to rebroadcast based on the local information available at the node or on the information carried by the messages exchanged between nodes. Conceptually, the approach seeks to balance the redundancy of paths required for robustness with the need to control excessive messaging that causes collisions and waste of bandwidth.

PROBLEM DESCRIPTION

The motivating scenario for this work is the creation of ad-hoc smart environments or smart structures on-demand using large numbers of very simple low-energy sensor elements with rudimentary computation and communication capabilities. The system communicates messages between arbitrary points in the network. A user, possibly operating within the system, interacts with the system by establishing a connection with any node anywhere in the sensor network. The main goal is to look at issues that would arise for truly large networks. We focus primarily on the networking algorithms aspect rather than on issues such as hardware, signal processing and communication.

The key attributes of our system are:

1. The user has minimal control over the distribution of the sensors beyond selecting the general region of their deployment.

2. The direct communication range of individual sensors is much smaller than the size of the deployment region. The implication of this constraint is that all algorithms must be able to function primarily with local data, and should not have to rely on global information.

3. All message addressing in the system is purely geographical, i.e., destinations are identified only by their location. We assume that the nodes do a Voronoi tessellation of the space and each node knows what region it is responsible for.

4. The nodes have minimal computational capability and very low energy. This means that there is no room for complex signal processing or optimization algorithms at the node level.

5. Nodes have a reasonably accurate estimate of their geographical location (via GPS or localization), and are well synchronized.

6. Nodes have a relatively high failure rate, and are often temporarily out of commission due to energy limitations or physical factors. These failures should not appreciably affect performance and should be invisible to the user.

7. All data communication in the system is broadcast. The only loop control in the system is based on the principle that no message will be re-broadcast more than once by the same node. At the origin, each packet is given a unique identifier, and each node receiving the packet caches this identifier. A message, whose identifier is found in the cache will not be broadcast on the assumption that it was either re-broadcast or rejected for re-broadcast earlier.

To this basic system, we add simple heuristics that allow each node to decide whether to re-broadcast (forward) a received message or not depending on received data, its destination, and the local information available at the node.

We use a medium-access protocol called *CSMA with mini-backoff (CSMA-mb)*, that is suitable for intelligent broadcast [1, 12]. It is based on slotted CSMA, but uses a two-level contention scheme. The channel is divided into major slots and each major slot is divided further into mini-slots. The size of a mini-slot is equal to the maximum 1-hop propagation delay a in the system. The length of a major slot is an integral multiple of the mini-slot. The first m mini-slots in a major slot are reserved for contention while the rest are set aside for data. Nodes are allowed to contend only at major slot boundaries. Nodes sense the channel in the first mini-slot of every major slot. If a node senses a busy channel at the beginning of a major slot, it backs-off to the next major slot (persistent protocol). If the node senses the channel to be free, it sets a random *mini-backoff*, the value of which lies in the range $[1, m]$. Each one of the nodes that contend at the beginning of a major slot would thus set a mini-backoff. The node senses the carrier again when its mini-backoff expires and starts transmitting in the event of an idle channel. Thus, the node that sets the smallest mini-backoff in a neighborhood wins the major slot. All nodes that set a higher back-off value would then sense the channel busy when their respective mini-backoff expires and hence back-off to the next major slot. Collision in CSMA-mb is a possibility only when two or more neighboring nodes with messages to send in the same slot also set the same mini-backoff value.

If standard Aloha [2] is *impolite* and pure CSMA is *impatient*, CSMA-mb can be considered a *cautious* protocol: it is focused on avoiding collisions even among messages that have equal "rights" to a slot. In slotted CSMA, nodes transmit immediately after sensing a free channel and neighborhood collisions are common, but the queues at the nodes are short due to the incautious nature of the protocol. CSMA-mb trades-off wait time at the nodes for improved delivery rate [1]. Also, since collisions can be reduced by increasing the contention period, CSMA-mb provides a flexible framework to achieve a desirable trade-off between message delivery and latency.

INTELLIGENT BROADCAST SCHEME

Self-organized Contention and Routing in Intelligent Broadcast Environments (SCRIBE) is an approach we propose to work around the broadcast storm problem by adaptively controlling the redundancy of message paths. The key idea behind SCRIBE is *intelligent dequeueing* of messages by nodes between the message source and destination, thus reducing collisions created by unnecessary messages. The forwarding of messages in this scheme is based on need rather than on some static heuristic such as local density of nodes or the

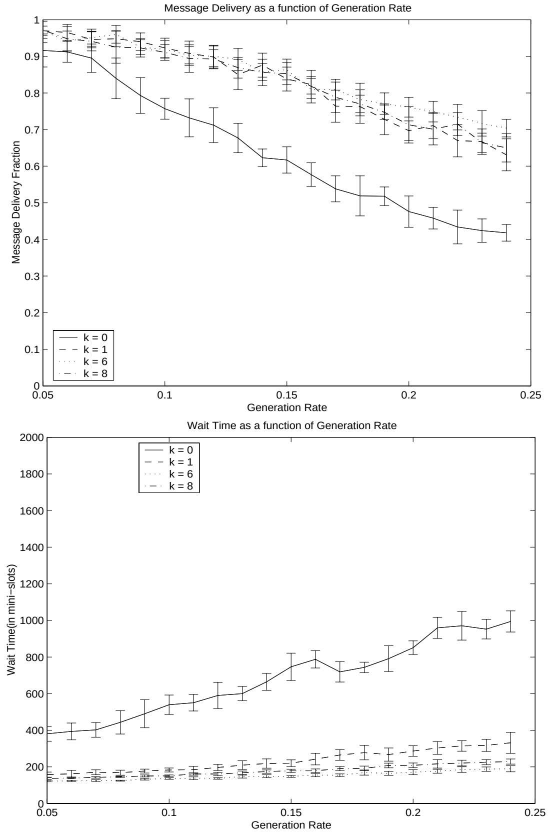

Figure 1: Effect of intelligently dequeueing the messages on message delivery and wait times of flooding under infinite energy conditions.

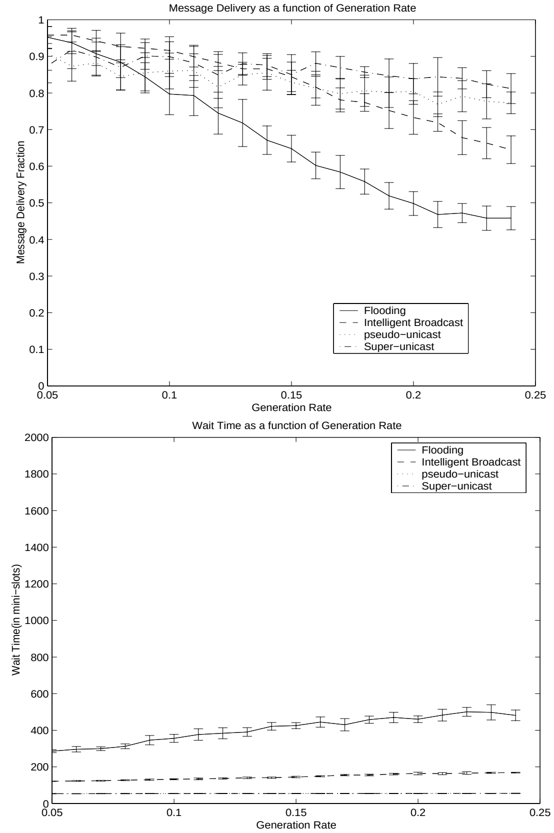

Figure 2: Message Delivery and Wait times under infinite energy conditions for 500 node networks. All data is averaged over 10 independent networks.

message's hop count [14]. This forwarding need is estimated as follows.

An intermediate node receiving a data message determines which k of its neighbors provide the best progress towards the message's destination, *rating* them according to their geographical proximity to the destination, and encoding this information in the forwarded data message. Nodes receiving the message acknowledge (ACK) receiving it, but in the order of their rating, i.e., a node with a lower rating acknowledges the data message only if it does not hear an ACK from a higher rated neighbor. However, it does forward the message. Unrated nodes in the neighborhood simply drop the message from their queues upon receiving an ACK for that message from a rated neighbor. Thus, the inescapable "overhearing" of the ACKs by nodes in the region limits the need for multiple ACKs, and allows dequeueing by many potential forwarders by informing them that better-placed nodes have already heard the message. If the original forwarding node hears no ACKs in a set time, it queues the message for retransmission, thus ensuring reliable delivery. We see this as achieving the *best available redundancy* of paths throughout the network — enough to ensure forward progress, but no more.

The main requirement for this scheme is that each node know the (approximate) geographical coordinates of its neighbors. This information is exchanged by the nodes through dedicated messages called *control* or *hello* messages during an initial set up period. This scheme is similar to an approach used to improve the performance of Time Spread Multiple Access (TSMA) protocols [9].

ACK messages are, of course, a common feature in ad-hoc networks, and are sometimes carried by a separate channel. However, we provide explicit support for ACKs by adding special mini-slots at the end of each CSMA-mb slot. Within these mini-slots, the contention for ACK messages works just like data messages, except that the mini-backoff set by each node is proportional to its rating. A hidden terminal collision of ACK messages is still possible in this situation, but is very unlikely because the nodes contending for the ACK channel are typically to the same side of the transmitting node, and are able to hear each other.

RESULTS AND DISCUSSION

We considered systems where nodes are distributed randomly with a uniform density in a square environment. Each node communicates only over a limited radius, R, chosen to almost guarantee full connectivity [13]. Messages were generated by a Poisson process with mean rate λ between 0.05 and 0.24, which ensured that the network operated in a region far from overload. Using a discrete-event simulator, SCRIBE was compared with three idealized protocols:

1. *Pure Flooding*: A non-destination node upon receiving a message, blindly floods the message to the node's neighbors.

2. *Pseudo-Unicast*: This is a "worst-case unicast", where each non-destination node, upon receiving a message, unicasts it to the neighbor providing the greatest progress towards the destination [13]. Nodes do not employ any channel-reservation scheme [7], and thus collisions are prevalent. However, traffic is reduced over flooding because most nodes hearing a message do not forward it. Pseudo-unicast is like broadcast without redundancy, and provides a way to study the value of redundancy.

3. *Super-Unicast*: This is an idealized "best-case unicast" where a transmission from any node A to its best placed neighbor B is always successful, provided B is in commission, i.e., collisions are ignored.

Using these two idealized types of unicast allowed us to compare the performance of SCRIBE with the whole range of unicast protocols without implementing them explicitly.

The performance of the protocols was evaluated under an ideal scenario with no failures (*infinite energy*) and under a *random failures scenario*, where the nodes were divided into *stable* nodes, which were not subject to failure, and *normal* nodes, that failed with probability P_f in each slot. Two metrics were used to quantify performance: 1) *Mean Message Delivery Rate*, defined as the ratio of total messages generated to the actual messages successfully received by the intended final destination; 2) *Mean Wait Time*, which is the average time a message waits in a node's queue before it is transmitted.

Figure 1 shows performance curves for various values of k in the infinite energy scenario. A marked difference in delivery and wait times can be observed when k goes from 0 (pure flooding) to 1 — intelligent broadcast with minimum guaranteed redundancy. For $k \geq 1$, $k = 6$ gives the best delivery rate and wait time, though the difference is only marginal.

Figure 2 shows performance curves for all protocols in the infinite energy case for 500-node networks. The two unicast protocols perform somewhat better than IB, presumably because traffic reduction due to unicast more than makes up for lack of path redundancy. However, results with networks of smaller and larger sizes (not shown) indicate that IB is increasingly better as system size grows and necessitates longer paths with more hops, enhancing the benefits of redundancy.

Figure 3 shows the performance of 100-node systems with 50 stable nodes as a function of λ and two different values of P_f (0.2 and 0.8). Figure 4 shows the results for 100-node systems as a function of the fraction of stable stable nodes, with $P_f = 0.5$ and two different λ values ($\lambda = 0.1$ and 0.2). It is apparent from the figures that, as the probability of failure, P_f, increases, the broadcast protocols with their inherent parallelism, perform much better than unicast-based protocols. *Intelligent broadcast* combines the redundancy of flooding with the specificity of unicast in an intelligent fashion. In spite of managing a superior delivery under failures, this scheme maintains a wait time comparable to that of unicast-based protocols. In addition, Figure 3 shows that the unicast-based protocols perform very poorly when more than half of the nodes in the system are unreliable.

Figure 5 give the results for 500 node systems subjected to failures. A dramatic difference in the delivery rate can be observed between unicast and broadcast protocols, demonstrating the increasing utility of IB in large networks with unreliable nodes. It should be noted that the advantages of broadcast-based redundancy obtain mainly in the absence of explicit handshake schemes such as MACA [7]. However, as nodes become smaller and networks larger with extremely high hop counts for most messages, channel reservation mechanisms will be increasingly impractical. Our work is oriented towards those situations, and demonstrates the value of simple, redundancy-based communication rather than sophisticated protocols. Of course, the work presented can be improved further, e.g., by using a more traffic-dependent ACK scheme, a more realistic failure model, and imperfect localization. These issues will be pursued in future research. More detailed results for this and related work can be found at *http://www.ececs.uc.edu/∼aminai/*.

References

[1] R. Arumugam , V. Subramanian, and A.A. Minai. "Intelligent Broadcast for Large-Scale Sensor Networks", To appear in *Proc. Int. Conf. on Complex Systems (ICCS2002), June 2002.*

[2] N. Abramson. , The aloha system - another alternative for computer communications. *In 1970 Fall Joint Computer Conference, AFIPS Conference Proceedings, volume 37, pages 281-285. Montvale, NJ: AFIPS Press, 1970.*

[3] D. Estrin, R. Govindan, and J. Heidemann. Scalable coordination in sensor networks. *Technical Report 99-692, University of Southern California, January 1999.*

[4] F. Ye. GRAdient Broadcast: A Robust, Long-lived Sensor Network. *http://irl.cs.ucla.edu/papers/grab-tech-report.ps*

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *In Proc. IEEE Hawaii Int. Conf. on System Sciences, January 2000, pp. 4–7.*

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *In Mobile Computing and Networking, 2000.*

[7] P. Karn. MACA - a new channel access method for packet radio. *ARRL/CRRL Amateur Radio 9th Computer Networking Conference 1990.*

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

[9] R. Krishnan and J.P.G. Sterbenz. An Evaluation of the TSMA Protocol as a Control Channel. *BBN Technical Memorandum No. 1279 Job No. 11982000 26 April 2000*

[10] J. Kulik, W. R. Heinzelman, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *In Proc. 5th Intnl. Conf. Mobile Computing and Networking, pages 174–185, 1999.*

[11] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu. The Broadcast Storm Problem in Mobile Ad-Hoc Network. *In Int'l Conf. on Mobile Computing and Networking(MOBICOM), Aug. 1999.*

[12] V. Subramanian, R. Arumugam, and A.A. Minai. "Self-Organization of Geographical Routing in Large-Scale Sensor Networks", To appear in *Proc. Int. Conf. on Complex Systems (ICCS2002), June 2002.*

[13] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. on Comm., 32(3):246–257, March 1984.*

[14] Y.-C. Tseng, S.-Y. Ni, E.-Y. Shih. Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad-Hoc Network. *In Int'l Conference on Distributed Computing Systems(ICDCS-21), April. 2001.*

[15] Y. Xu, J. Heidemann and D. Estrin. Adaptive Energy-Conserving Routing for Multihop Ad-Hoc Networks. *Research Report 527, USC/Information Sciences Institute, October 2000.*

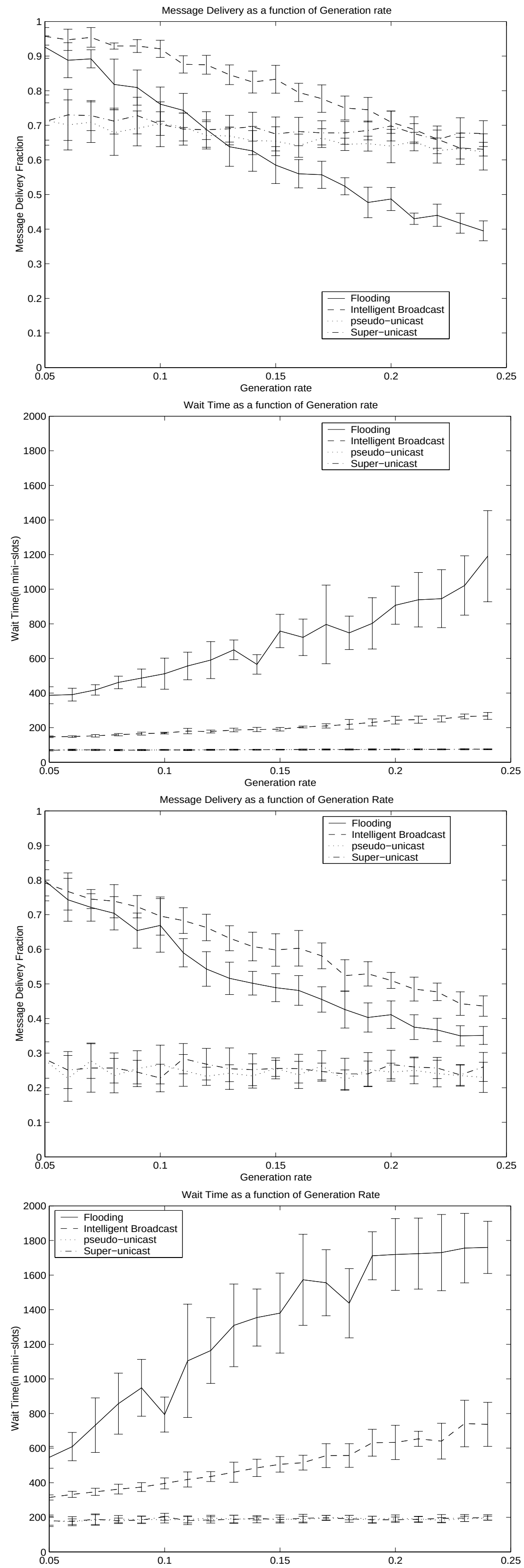

Figure 3: Message Delivery and Wait times under random failures with $P_f = 0, 2$ (top two graphs) and $P_f = 0.8$ (bottom two graphs)as a function of generation rate for a 100 node network.

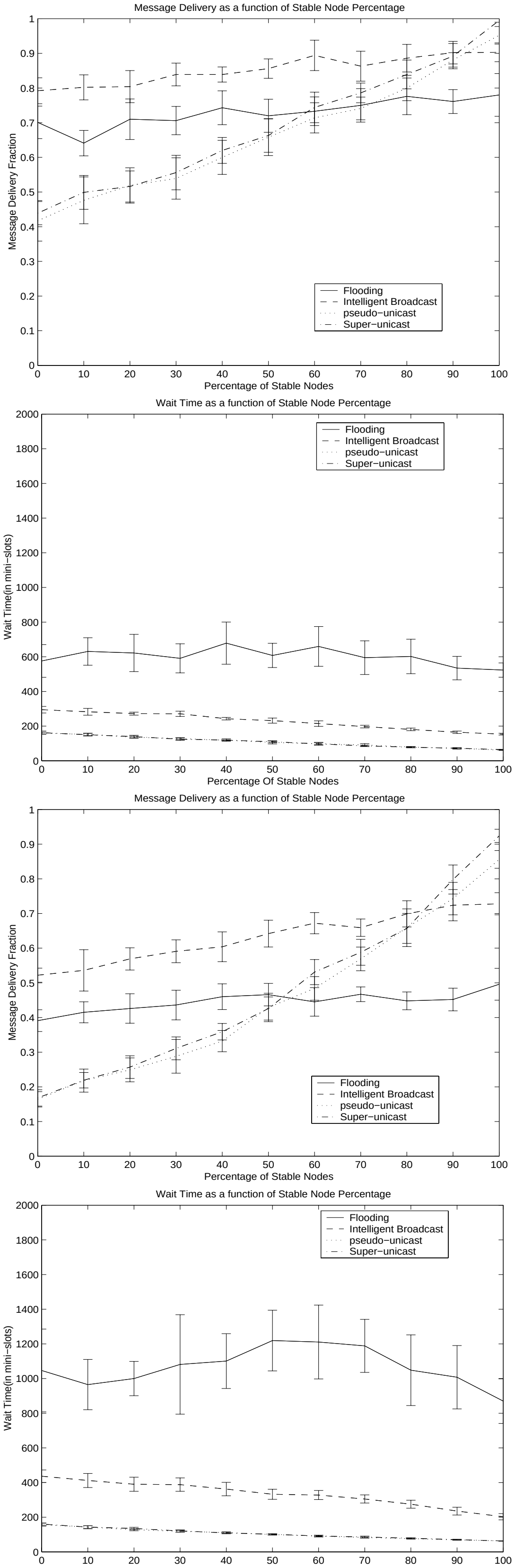

Figure 4: Message Delivery and Wait times for Flooding, Intelligent broadcast, pseudo-unicast and Super-unicast under random failures over percentage of stable nodes for a 100 node network.

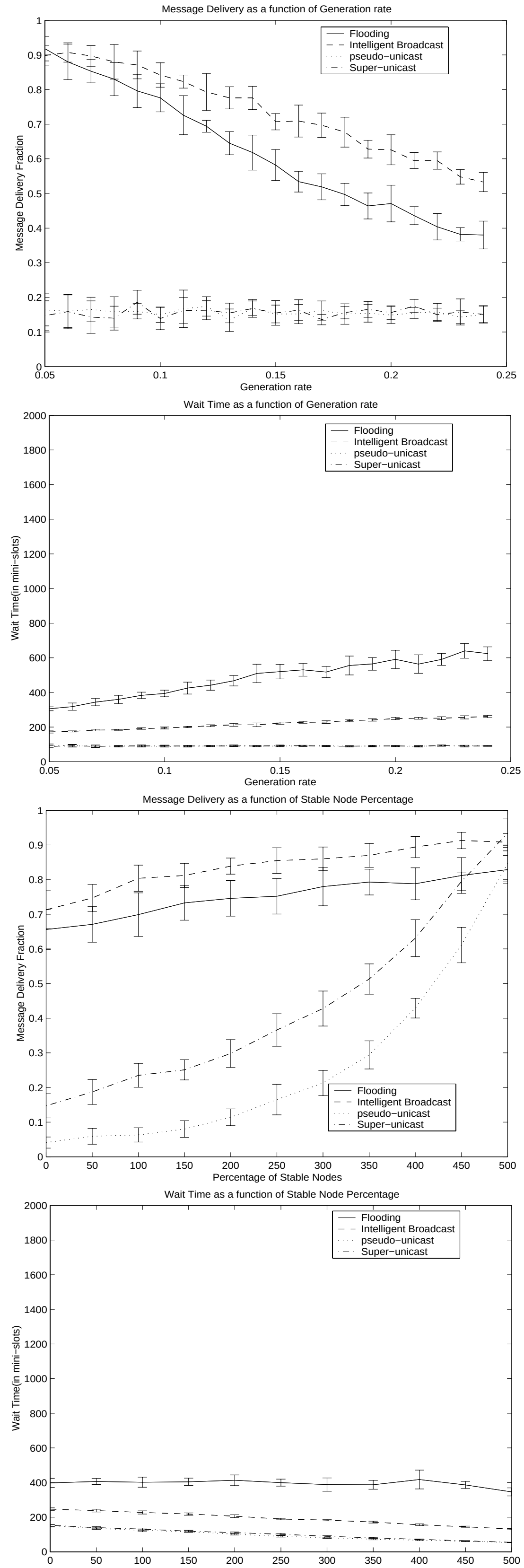

Figure 5: Message Delivery and Wait times for Flooding, Intelligent broadcast, pseudo-unicast and Super-unicast under random failures for a 500 node network.