Intelligent Broadcast For Large-Scale Sensor Networks

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

With advances in miniaturization, wireless communication, and the theory of self-organizing systems, it has become possible to consider scenarios where a very large number of networkable sensors are deployed randomly over an extended environment and organize themselves into a network. Such networks — which we term *large-scale sensor networks (LSSN's)* — can be useful in many situations, including military surveillance, environmental monitoring, disaster relief, etc. The idea is that, by deploying an LSSN, an extended environment can be rendered observable for an external user (e.g., a monitoring station) or for users within the system (e.g., persons walking around with palm-sized devices). Unlike custom-designed networks, these randomly deployed networks need no pre-design and configure themselves through a process of self-organization. The sensor nodes themselves are typically anonymous, and information is addressed by location or attribute rather than by node ID. This approach provides several advantages, including: 1) Scalability; 2) Robustness; 3) Flexibility; 4) Expandability; and 5) Versatility. Indeed, this abstraction is implicit in such ideas as smart paint, smart dust, and smart matter.

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 each node has 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. In particular, we are interested in obtaining reliable communication at the system level from simple, unreliable nodes.

1 Introduction

Wireless networks that operate without fixed infrastructure are called ad-hoc networks, and are a very active focus of research by the wireless community. However, most of the work focuses on networks with tens or hundreds of nodes, where most message paths are only a few hops long. All data messages in such a system are unicast, i.e., they are between specific pairs of nodes. There are two major formulations for this. In some message routing algorithms, a path discovery process is used to first find a route between the source and destination nodes (or locations), and the message is then sent along this path [7, 8, 4]. This is clearly a top-down approach with limited scalability. Other routing protocols use next-hop routing, where each node, knowing the destination of an incoming message, only determines the next node to forward the message to [3]. These protocols scale much better, but at the cost of maintaining and updating extensive amounts of information about network topology. This can be expensive in terms of energy, and can often lead to problems if the individual nodes are unreliable, causing broken links and lost messages. From a complex systems viewpoint, the problem with unicast-based next-hop methods is that they do not exploit the inherent parallelism of the system to achieve robustness. This is the issue we consider in our research.

Rather than using directed unicast between nodes, we study the possibilities of broadcast. In the simplest case, this corresponds to flooding, where every message received by a non-destination node is "flooded" to all the node's neighbors. While this is a simple approach, it is extremely wasteful of bandwidth and creates a lot of collisions — the simultaneous use of the wireless channel by multiple messages, all of which are lost as a consequence. To overcome the problems of flooding while retaining its inherent parallelism, we explore the method of intelligent broadcast. In this approach, each node receiving a message decides whether to re-broadcast it to all its neighbors or to ignore it. Note that the decision does not involve selecting *which* neighbor the message is forwarded to, but only *whether* to forward the message. The latter is a much simpler decision, and can be made on the basis of the information carried by the message in combination with that available within the potential forwarding node. This approach leads to a self-organized communication process where local decisions by the nodes produce global availability of information.

In this paper, we present a well-developed paradigm for random LSSN's, including a model for the nodes and viable broadcast-based protocols for channel access and network organization. We evaluate the performance of the network in the case of simple flooding, and then study the effect of a simple decision heuristic that allows nodes to limit message re-broadcast based on how many hops the message has already traveled. We show that this heuristic leads to a dramatic improvement in performance, making the broadcast-based system a viable — and more robust — alternative to more complicated systems under some conditions.

2 Medium Access Protocols

Traditional networks make use of point-to-point channels, in which interference-free communication is established between an ordered pair of users. But, under scenarios where such channels are not economical or are not available, broadcast channels are used. When nodes communicate through a broadcast channel, a single transmission by a node is heard by all nodes within the transmitting node's radius of communication. In essence, the success of a transmission between a pair of nodes is no longer independent of other transmissions: The channel becomes a shared resource whose allocation is critical for network operation. Schemes for channel allocation are known as *medium access protocols*. Medium access protocols can be broadly classified as *conflict-free* and *contention* protocols. Conflict-free protocols ensure successful transmission at all times. This can be achieved by allocating the channel to the users either statically or dynamically. In contention based schemes, a transmitting user is not guaranteed to be successful. The protocol must prescribe a way to resolve conflicts once they occur. An example is the *Aloha* protocol, which was the first random access protocol implemented. In aloha, a newly generated packet is transmitted immediately hoping for no interference from other users. Another contention-based protocol, CSMA, is described below.

2.1 Carrier Sense Multiple Access (CSMA)

In CSMA [6], nodes in the network sense the channel prior to transmitting packets. If the channel is sensed to be busy, the sensing node refrains from transmitting, avoiding a collision in the process. The node reschedules the packet for a later time. This delay is termed the *back-off*. If the channel is sensed to be idle, the node transmits its packet. In *slotted CSMA*, the channel is divided into mini-slots, the width of a mini-slot being equal to the maximum propagation delay in the system. Nodes using slotted CSMA are restricted to start transmissions only at mini-slot boundaries.

Even though slotted CSMA improves over the unslotted version by synchronizing the nodes and thus reducing the length of the unsuccessful transmission periods, it does little to improve the quality of channel access. Nodes with packets ready for transmission in a neighborhood, upon sensing a busy channel, back-off for a random duration (depending upon the back-off strategy). Our argument is that by backing-off to a later time, the nodes are not preventing but just deferring the collisions to a later time. Since nodes back-off to integral multiples of the slot, each slot is subject to collision among messages, generated at various times in the past, that happen to choose the same back-off point. We call this the problem of *colliding backoffs*, and it is an inherent disadvantage of slotting.

Consider another case in a multi-hop network when the channel in a neighborhood has been idle for a length of time. A node, upon generating a packet or receiving one for forwarding, would sense the channel and would transmit the packet if the channel was idle. In a broadcast CSMA channel, all the neighbors of this node would receive the packet and hence sense the carrier at the next mini-slot boundary. In all probability, many of the nodes would sense the channel free and hence transmit at the same instant. Since some of these nodes are also neighbors of each other, this would result in collision of the packets. Thus, in this protocol, a successful transmission would trigger a spate of collisions in a relatively calm neighborhood. In the next section, we propose a novel channel access scheme that solves this problem by allowing each node a fair chance to earn a slot.

2.2 CSMA with Mini-Backoff (CSMA-mb)

To overcome the problems with slotted CSMA in the context of multi-hop networks, we propose a novel persistent scheme termed *CSMA with mini-backoff (CSMA-mb)*. CSMA-mb is built over the slotted CSMA protocol and shares the same philosophy with regard to the maximum propagation delay in the system, i.e., in a multi-hop network, since the communication is localized (between one-hop neighbors), the maximum propagation delay, a, between the nodes is assumed to be not too large. In pure slotted CSMA, a collision could occur either between messages contending in a slot for the first time or among messages generated at various times in the past that happen to choose the same back-off point. In CSMA-mb, we give these messages a further opportunity to resolve their contentions by allowing them to backoff yet another time, albeit for a relatively smaller duration called mini-backoff, and hence avoid a collision. The idea behind CSMA-mb is to provide a method whereby contention can be resolved *within* a slot — thus greatly reducing the chances for collision due to colliding backoffs. The CSMA-mb protocol is described below.

- In CSMA-mb, we divide the channel into major slots and further divide each major slot into mini-slots. The structure of a major slot in CSMA-mb is shown in Fig. 1.

- The size of a mini-slot is equal to the maximum 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.

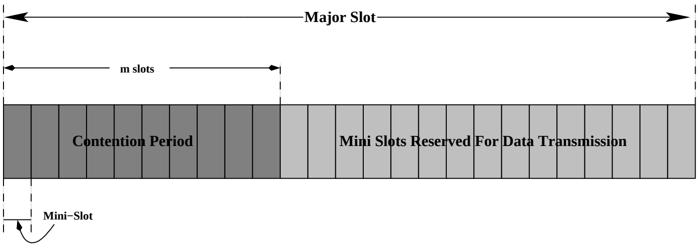

Figure 1: Data structure of a slot in CSMA-mb

- 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 as a consequence of the persistent nature of the protocol.

- If a node senses the channel free, it does not transmit right away. Rather, it sets a random mini-backoff, the value of which lies in the range [1, m], where m is the maximum number of mini-slots reserved for contention. Thus, each one of the nodes that contend at the beginning of a major slot would set a mini-backoff.

- A node then senses the carrier again when the 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 nodes with messages to send in the same slot set the same mini-backoff value. Thus, the performance of the system can be improved further by increasing the length of the *contention period*.

CSMA-mb is explained with an example scenario in Fig. 2. Assume that neighbors p, q, r and s have packets to send at time t and hence sense the carrier. Assuming that the nodes sensed a free carrier, each of the nodes would set a mini-backoff. Suppose, p, q, r and s, backed-off to times t_1, t_2, t_3 and t_4 respectively such that $t_2 < t_1 < t_4 < t_3$. Thus, q would sense the carrier free when its mini-backoff expires and would start transmission immediately afterwards. Nodes p, r and s would sense a busy channel when their respective mini-backoffs expire and hence would back-off to the next major slot. If pure slotted CSMA had been used for channel access, the packets from p, q, r and s would have collided.

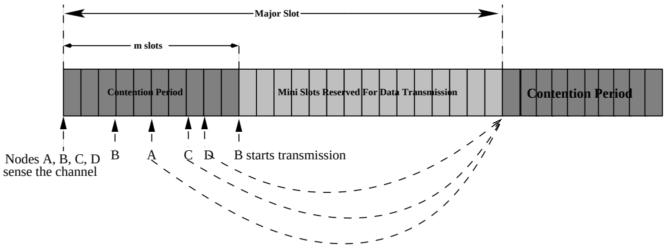

Figure 2: Channel Access in CSMA-mb

In conclusion, if Aloha is *impolite*, pure CSMA is *impatient*. In a similar vein, CSMA-mb can be characterized as a *cautious* protocol: it is focussed on avoiding collisions even among messages that have equal "rights" to a slot. From a complex systems viewpoint, CSMA-mb is attractive because, the efficiency in the messaging process is achieved by resolving the contentions and consequent collisions locally.

Next, we describe a simple heuristic that tries to improve network performance further by reducing collisions. It does so by controlling undesirable redundancy of message paths in the system.

3 Hop limit for messages

Broadcast-based communication is wasteful of resources, but it also obviates the need for gathering, storing and frequently updating massive amounts of information about a vast network. As these networks grow very large, keeping track of routes or link states will not be possible, but broadcast will always work. Also, very large-scale networks will need to use extremely cheap and, therefore, unreliable nodes. Frequent random failures of nodes in the network would make precisely directed communication inefficient unless nodes monitored each other very frequently, which would be too expensive. Using intelligent broadcast is a way to use the inherent parallelism of broadcast and the massive distributedness of the system to ensure communication.

Intelligence in broadcast can be derived either from the local information available at a node or from the information carried by the data messages. In addition, nodes can also exchange information by dedicated messages, called control or *hello* messages. These messages are usually exchanged by the nodes at periodic intervals for gathering and updating information about their local neighborhoods.

In this work, we study the effect of nodes exchanging information about the state of a message on the performance of flooding. A node, upon generating a message, assigns an upper limit to the number of hops the message can traverse in the system. This maximum limit on the hops is generally know as *hop limit*. The notion of hop limit has been explored by researchers in the past. In dynamic source routing [4], the initiator of the route request specifies the maximum number of hops over which the packet may be propagated. The packets are initially assigned a hop limit of one. If no route reply is received for this request, a new route request with the hop limit set to a predefined ”maximum” is sent out. In the store and flood protocol [2], a pre-defined higher limit on the number of hops over which a message may propagate is assigned.

Our contention is that, in an LSSN, some messages might be directed to a node only a few hops away while others might have to traverse the entire length of the system. Thus, assigning a predetermined maximum to the hop limit (based on the maximum number of hops possible in the system) would benefit only messages involved in end-to-end transmission. Besides, arriving at an optimum value for hop limit off-hand is a non-trivial problem. We believe, the problem is best left to the source of the message. In our system, the messages are assigned a hop limit proportional to the expected number of hops to the destination by the source node at generation. Hence, each message has a different hop limit depending upon the geographical distance it needs to traverse.

The hop limit assigned to a message is decremented at every hop it makes. A non-destination node, upon receiving a message for forwarding, rebroadcasts the message only if the hop limit has a non-zero value. The results of this simple heuristic on the performance of flooding are discussed in §5. The performance obtained in this way has proved to be as good as, if not better than, assigning a hop limit equal to the average hop length of successful messages in the system — a statistic that is difficult to calculate at the node level. The next section gives a brief description of the system.

4 System Description

The key attributes of the system we consider are:

1. The user has minimal control over the distribution of the sensors beyond selecting the general region of their deployment. The sensors are then randomly distributed over this geographical space.

2. The direct communication range of individual sensors is very small compared to the size of the deployment region. In general, any pair of sensors is separated by a large number of hops. 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. The sensors are anonymous and all addressing in the system is purely geographical, i.e., sensors are identified only by their location.

4. The sensors have minimal computational capability and limited energy. This means that there is no room for complex signal processing or optimization algorithms at the sensor level.

5 Simulation and Results

Nodes are randomly distributed with a uniform density in a non-discretized environment of linear size l. Each node can communicate only over a limited radius, R. We choose an R, which just about guarantees full connectivity for a given network size and density. Messages are generated according to a Poisson process with mean rate λ, with each message being assigned a unique identifier. Each node receiving a 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 earlier or rejected for re-broadcast earlier. This loop control in the system is based on the principle that no message will be re-broadcast more than once. Moreover, the message generation rate is chosen such that the network is not overloaded.

The network was simulated for $l = 50$ units, $\lambda = 0.05$ to 0.24, $R = 8$ and 3 units for 100 and 500 nodes respectively. The width of the major slot in CSMA-mb is 85, with $m = 30$ dedicated for contention. We chose two metrics to analyze the performance of flooding:

1. *Average Message Delivery:* The ratio of total messages generated to the number successfully received by the intended final destination.

2. *Average Wait Time:* The average time a message waits in a node's queue before being forwarded.

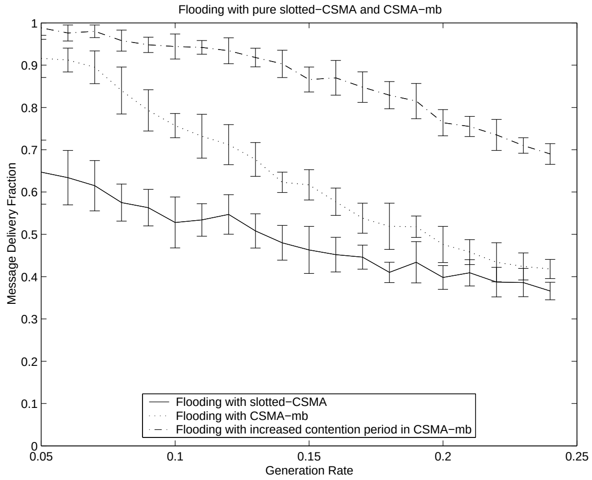

Figure 3: Slotted CSMA versus CSMA-mb:100 nodes

It is apparent from Fig. 3 that CSMA-mb is much superior to pure slotted CSMA in terms of delivery. In addition, Fig. 3 illustrates that by increasing the contention period for the nodes, a much higher delivery rate can be achieved.

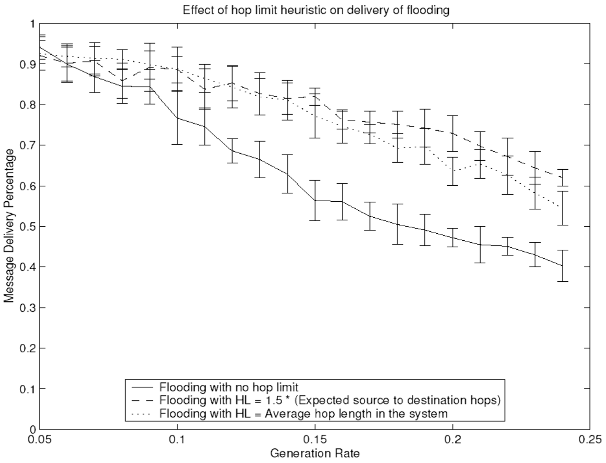

Figure 4: Effect of hop limit on delivery of flooding for 100 nodes

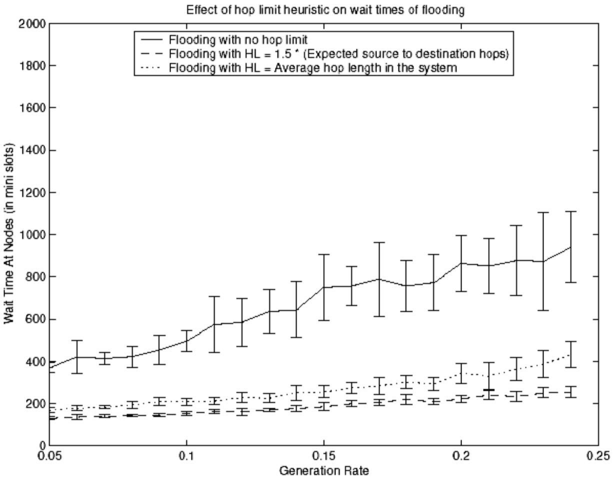

Figure 5: Effect of hop limit on wait times of flooding for 100 nodes

Fig. 4 and Fig. 7 show that a simple heuristic like hop limit can improve the performance of flooding dramatically. Moreover, our contention that assigning a hop limit based on the geographical separation of the source and destination of a message is a more intelligent approach than using a predefined maximum value appears to be correct. Very importantly, we have been ale to demonstrate that a *locally computable* hop limit heuristic can match — or even surpass — the performance obtained from the approach of using the average hop length of

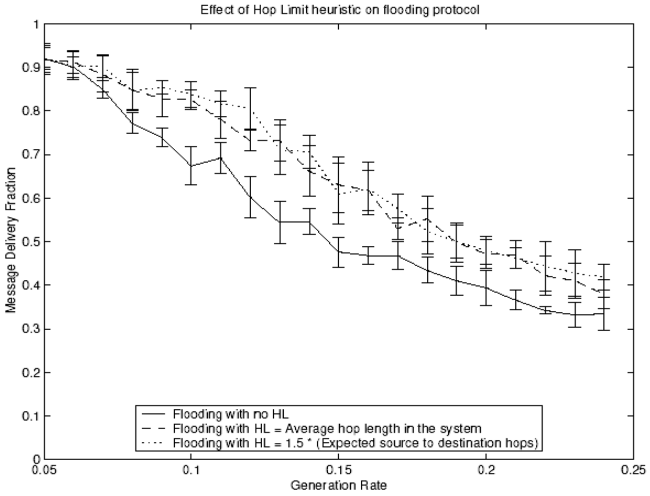

Figure 6: Effect of hop limit on delivery of flooding for 500 nodes

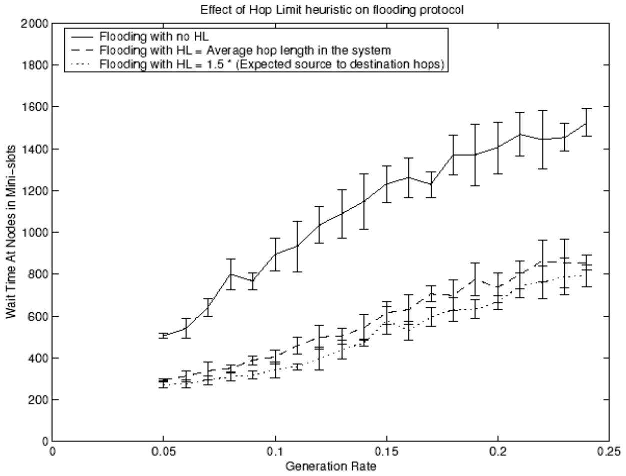

Figure 7: Effect of hop limit on wait times of flooding for 500 nodes

successful messages, which can only be calculated globally over the network.

6 Conclusion and Discussion

In this paper, we introduced CSMA-mb (Carrier Sense Multiple Access with mini-backoff), a persistent medium access protocol for broadcast-based networks.

CSMA-mb gives the nodes accessing the channel a fair chance to earn a slot by providing dedicated slots for contention. CSMA-mb trades-off wait time at a node to achieve superior delivery. Thus, messages in CSMA-mb reach the destination with increased latencies. We also presented a new heuristic for hop limit in the system and analyzed the effect of this heuristic on performance of flooding. Our method represents a practical approach to communication in large-scale sensor networks. The local nature of the method makes it much more scalable than traditional global methods, and, therefore, more suitable for self-organized complex adaptive networks as discussed in[1].

Bibliography

[1] Arumugam, Rajkumar "SCRIBE: Self-Organized Contention and Routing in Broadcast Environments", *MS Thesis*, University of Cincinnati, May, 2002

[2] Hagino Hiroaki, Takahiro Hara, Masahiko Tsukamoto, Shojiro Nishio. "Store and Flood: A Packet Routing Protocol for Frequently Changing Topology with AD-Hoc Networks" *IPSJ JOURNAL* Abstract Vol.41 No.09 - 010.

[3] Jain, Rahul, Anuj Puri, and Raja Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks", *Technical Report M99/69*, University of California, Berkeley Dec. 1999

[4] Johnson David B., David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", *Mobile Computing*, Kluwer (1996), 153-181.

[5] Kleinrock, Leonard, J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number", *Proceedings of the IEEE National Telecommunications Conference*, (Dec. 1978), 4.3.1–4.3.5.

[6] Kleinrock, Leonard, F.A. Tobagi, "Packet switching in radio channels: Part–I - carrier sense multiple access modes and their throughput-delay characteristics", *IEEE Transactions in Communications*, COM 23 (12)(1975), 1400–1416.

[7] Perkins, Charles E., and Elizabeth M. Royer, "Ad-hoc On-Demand Distance Vector Routing", *Proc. 2nd IEEE Wkshp. Mobile Comp. Sys. and Apps.*, (Feb. 1999), 90–100.

[8] Perkins, Charles E., and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Comp. Commun. Rev.*, (Oct. 1994), 234–244.