UNIVERSITY OF CINCINNATI

_____, 20 ____

I, _____,
hereby submit this as part of the requirements for the
degree of:

in:

It is entitled:

Approved by:

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

A thesis submitted to the

Division of Graduate Studies and Research
of the University of Cincinnati

in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in the Department of
Electrical and Computer Engineering and Computer
Science

of the College of Engineering

by

Rajkumar Arumugam

B.E., University Of Madras, 1999

Thesis Advisor and Committee Chair: Dr. Ali Minai

Dedication

*In the loving memory of
ammachi
and
periamma*

Abstract

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 or other nodes in the system).

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.

In this thesis, 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 explore methods for intelligent broadcast where the efficiency in the messaging process is created by a node's decision to rebroadcast a message. This allows the communication process to utilize redundant paths without unnecessary messaging. We evaluate the performance of the network as the degree of sophistication of the underlying communication mechanism grows. We explore a range of methods, starting with simple flooding and going up to unicast communication. We also study the robustness of each of these protocols when subjected to random failures.

Acknowledgements

First of all, I would like to thank Dr.Minai for being such a great human being and an advisor par excellence. His support and encouragement for the past two years have been unconditional and wholesome. This work would not have been possible without his colossal patience and impeccable guidance.

I would like to thank Dr.Das and Dr.Caffery for taking time off their busy schedules to be on the defense committee. I would also like to take this opportunity to thank my lab-mate and friend Xiao fu for the numerous, often comical, Sino-Indo wars at 829 ERC and his unquestionable math fundas. Needless to say India came out trumps on most of the occasions!

I would have got it all wrong if I do not thank Vinod for being more than a friend to me. I would not be here if not for him, thanks Iyer! Thanks to Beri, Ansu and Sandy for the countless phone calls. Thanks also goes to Divram for his unabated words of wisdom and Bernie for making life at UC so much easier. Many thanks to Shyam for being such a great source of inspiration. Special thanks to appa, amma, Arun and Juju for always being there for me. And UC, I cannot thank enough!

Contents

1	Introduction	1
1.1	Ad-hoc Sensor Networks	1
1.2	Issues in Sensor Networks	2
1.3	Organization	7
2	Motivation and Background	8
2.1	Single-hop and Multi-hop Networks	9
2.2	Goals and Motivation	12
2.3	Medium Access Protocols (MAC)	15
2.4	The Broadcast Storm Problem(BSP)	25
2.5	Review of Data Handling in Sensor Networks	31
3	Methods	48
3.1	Problem Description	48
3.2	Overall Approach	50
3.3	An Improved Scheme for Medium Access	51
3.3.1	CSMA with Mini-Backoff (CSMA-mb)	53
3.3.2	Analytical Model	57
3.4	Intelligent Broadcast Scheme	69
4	Results and Discussion	79
4.1	Simulator Description	79
4.2	Slotted CSMA Versus CSMA-mb	88
4.3	Effect of intelligent dequeuing on performance of flooding	91
4.4	Protocols Comparison - Infinite Energy Conditions	101
4.5	Protocols Comparison - Random Failures	107
4.6	Hop Limit heuristic	126
4.7	Summary of the Results	133

5	Summary and Conclusion	136
5.1	Summary	136
5.2	Conclusion and Future Work	140

List of Figures

2.1	Hidden terminal scenario	11
2.2	Classification of Multiple Access Protocols	17
2.3	Unsuccessful Transmission Period in Unslotted CSMA	21
2.4	Slotted Nonpersistent CSMA Packet Timing	22
2.5	Additional area that benefits from a rebroadcast	26
3.1	Data structure of a slot in CSMA-mb	54
3.2	Channel Access in CSMA-mb	56
3.3	X to Y communication	59
3.4	Area of the Chord	59
3.5	The vulnerable period for CSMA protocols in multi-hop networks	63
3.6	Analysis of hidden terminal collisions in CSMA-mb	64
3.7	Collision of acknowledgements	74
3.8	Major Slot with Mini-slots for Ack in CSMA-mb	75
4.1	Slotted CSMA versus CSMA-mb	90
4.2	Analytical model Versus Simulation	92
4.3	Delivery fraction of intelligent broadcast scheme: No failures	95
4.4	Wait times of intelligent broadcast scheme: No failures	96
4.5	Delivery fraction of intelligent broadcast scheme: Stable nodes = 50%, $P_f = 0.5$	97
4.6	Wait times of intelligent broadcast scheme: Stable nodes = 50%, $P_f = 0.5$	98
4.7	Delivery fraction of intelligent broadcast scheme: $\lambda = 0.1$, $P_f = 0.5$	99
4.8	Wait times of intelligent broadcast scheme: $\lambda = 0.1$, $P_f = 0.5$	100
4.9	Comparison of delivery for all protocols: 100 nodes, no failures	103
4.10	Comparison of wait times for all protocols: 100 nodes, no failures	104

4.11	Comparison of delivery for all protocols: 500 nodes, no failures	105
4.12	Comparison of wait times for all protocols: 500 nodes, no failures	106
4.13	Comparison of delivery for all protocols: 100 nodes, Percentage of Stable Nodes = 50%, $P_f = 0.2$	110
4.14	Comparison of wait times for all protocols: 100 nodes, Percentage of Stable Nodes = 50%, $P_f = 0.2$	111
4.15	Comparison of delivery for all protocols: 100 nodes, Percentage of Stable Nodes = 50%, $P_f = 0.5$	112
4.16	Comparison of wait times for all protocols: 100 nodes, Percentage of Stable Nodes = 50%, $P_f = 0.5$	113
4.17	Comparison of delivery for all protocols: 100 nodes, Percentage of Stable Nodes = 50%, $P_f = 0.8$	114
4.18	Comparison of wait times for all protocols: 100 nodes, Percentage of Stable Nodes = 50%, $P_f = 0.8$	115
4.19	Comparison of delivery for all protocols: 100 nodes, $\lambda = 0.1$, $P_f = 0.5$	116
4.20	Comparison of wait times for all protocols: 100 nodes, $\lambda = 0.1$, $P_f = 0.5$	117
4.21	Comparison of delivery for all protocols: 100 nodes, $\lambda = 0.2$, $P_f = 0.5$	118
4.22	Comparison of wait times for all protocols: 100 nodes, $\lambda = 0.2$, $P_f = 0.5$	119
4.23	Delivery of super-Unicast with retransmission as a parameter	120
4.24	Wait times of super-Unicast with retransmission as a parameter	121
4.25	Comparison of delivery for all protocols: 500 nodes, Fraction of Stable Nodes = 50%, $P_f = 0.5$	122
4.26	Comparison of wait times for all protocols: 500 nodes, Fraction of Stable Nodes = 50%, $P_f = 0.5$	123
4.27	Comparison of delivery for all protocols: 500 nodes, $\lambda = 0.1$, $P_f = 0.5$	124
4.28	Comparison of wait times for all protocols: 100 nodes, $\lambda = 0.1$, $P_f = 0.5$	125
4.29	Delivery for flooding with and without hop limit on messages	129
4.30	Wait times for flooding with and without hop limit on messages	130
4.31	Delivery for flooding with and without hop limit on messages	131
4.32	Wait times for flooding with and without hop limit on messages	132

Chapter 1

Introduction

Ad-hoc networks are wireless networks that do not rely on any pre-existing infrastructure or centralized control for communication. The nodes in such a network are not only sources/sinks, but also routers/switches, supporting routing and other core network functions. Ad-hoc networks often find application in scenarios where one cannot expect or afford to have a fixed infrastructure at all times, e.g. in case of a disaster relief.

1.1 Ad-hoc Sensor Networks

Autonomous sensor networks in which the nodes coordinate among themselves to achieve a larger sensing task are revolutionizing the whole process of information gathering and dissemination. Advances in processor, memory and radio technology have made possible the realization of cheap and small

nodes capable of sensing in large scales.

Ad-hoc smart environments can be created using large numbers of very simple, low-energy sensor elements with rudimentary computation and communication capabilities. The sensors could provide active real-time on-demand information to the mobile user(s) for purposes such as path planning, navigation, assessment, surveillance, entertainment, etc. or to a user querying from a safe remote location. A canonical scenario might be where sensors are dropped onto hostile terrain before sending in autonomous vehicles to search for targets or to navigate safely through the terrain.

1.2 Issues in Sensor Networks

Over the recent years, large scale networks of wireless sensors have become increasingly feasible. If deployed, they would enable monitoring and control capability for applications in many areas, e.g., transportation, health care, environmental studies, security, and defense. As an example scenario, consider a city ravaged by an earthquake. Sending rescue teams into fragile structures in the immediate aftermath of an earthquake, without assessing the extent and nature of the damage and the prevailing conditions, could prove disastrous and sometimes fatal. Instead, a large number of inexpen-

sive sensors could be dropped off from a plane circling around the damaged infrastructure. The plane would act as the central base station querying the deployed nodes. These sensor nodes would self-organize and form an intelligent network that could relay important information about the state of the hostile environment in their immediate neighborhood. The data thus obtained could be processed and used as the basis for further human intervention. Self-organized sensor networks thus differ fundamentally from conventional wireless ad-hoc networks. Some of the design issues are outlined below.

Node Capability: A sensor network is typically made up of a large number of simple, inexpensive and unreliable nodes. These nodes are only capable of limited computation and simple decision making. The limitations imposed by memory and bandwidth do not allow these nodes the luxury of maintaining complex tables for routing and communication. In addition to these constraints, nodes only have limited power, warranting a multi-hop topology in majority of the networks. The nodes are usually powered by batteries and may last less than a few days. Since the nodes are simple, self configuring and do not need constant monitoring, deploying a sensor network is relatively inexpensive.

Reliability: Harsh environments and inhospitable terrains pose severe threats to the sensor nodes. In addition, the sensor nodes are likely to be unreliable because they must be cheap, and are often out of commission due to energy limitations. The central philosophy of a sensor network is that *redundancy* in sensor nodes would eventually lead to a reliable and robust network. Since the operations are localized and distributed, there is no single point of failure. No single node, is important for meaningful functionality of the network as a whole. In short, robustness in monitoring coverage and data forwarding is achieved through redundancy.

Connectivity: The user would typically presume that the entire sensor network is connected. This may not always be true, in fact, because sensors have limited communication range, and are distributed randomly. The main difficulty is that, while the user can query sensors directly over a large area, most sensors can respond (and hence, be detected by the user) only through the network. Thus, a sensor may “hear” the user without “being heard from”. The existence of such “silent sensors” indicates that the overall sensor population does not form a connected network.

Coverage: A network can be fully connected but still provide very poor coverage. This is possible if the sensors are distributed with very high density in a local region, leaving the majority of the space uncovered. It should be possible to determine the geographical extent of coverage by the sensors that can be accessed, directly or indirectly, from the user's current position. The question is: How does the user query the system to determine coverage? There are also secondary issues of how the coverage information is represented in sensors and how it is communicated to the user.

Robustness: As these sensor nodes are deployed in the real world, there will be a need to protect the sensor network from various types of attacks. For e.g., a sensor network deployed in a forest to monitor forest fires would be prone to attack by animals that inhabit the neighborhood. Sensors deployed for military applications could come under enemy attack. As a result, a part of the network could be damaged completely. The sensor network should be robust to such kind of failures. The network should easily adapt to the failure of existing sensors and accommodate addition of new ones to replace them.

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.

In this work, we study the possibility of broadcast in large scale sensor networks. 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. We look at broadcast in the first place because it is simple, inexpensive and exploits the inherent parallelism of the system to achieve robustness. Moreover, broadcast is the only possible means of communication in embedded sensor networks, e.g., smart paint. However, broadcast by flooding is extremely wasteful of resources and results in a lot of collisions. 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 messaging process is created by a node's decision to rebroadcast based on the local information available at a node, or through the information carried by the messages exchanged between them.

1.3 Organization

Chapter 2 of this thesis gives an account of the current literature on broadcast based ad-hoc sensor networks. Chapter 3 discusses the medium access models and protocols employed and, Chapter 4 details the results obtained and the subsequent discussion. Chapter 5 presents the conclusion and scope for future work in this area.

Chapter 2

Motivation and Background

The requirements of a sensor network greatly differ from those for conventional wireless ad-hoc networks. Unlike the case of ad-hoc communication networks, sensor networks are usually designed and deployed to span a large geographical space. This warrants a large scale deployment of sensor nodes to guarantee appreciable coverage. Also, the sensor nodes are usually very simple and are capable of only a minimal amount of computation and communication. This is in addition to the limited memory and energy resources that are typically available in such networks. Due to the need for low cost and the harsh environments to which these nodes are subjected, the sensor nodes are highly susceptible to failures. Thus, specialized approaches are needed to make networks of these nodes useful and reliable. In Section 2.5, we review some of the approaches that have incorporated this and other more

futuristic visions of sensor networks in their design. In the next section we give a brief description of single-hop and multi-hop networks.

2.1 Single-hop and Multi-hop Networks

In single-hop networks, all communication are one hop. That is, any node in the network can directly communicate with any other node in the network. The farther the distance between the nodes, the larger is the propagation delay between them. In contrast, communication is localized in multi-hop networks with intermediate nodes acting as routers for packets between distant nodes. Thus the propagation delay between any two nodes exchanging messages is much smaller in multi-hop networks when compared to single-hop networks because the nodes exchanging messages are within direct transmission range of each other in the former. Generally speaking, multi-hop networks have at least three main advantages over single-hop networks:

1. **Adaptability:** By deploying a multi-hop network in complex environments, packets can be routed around obstructions or inaccessible areas.
2. **Spatial reuse:** Communication in different parts of the network can proceed simultaneously since the messages are confined to the regions.

3. **Energy:** Multiple small radii transmissions require less energy than a single large transmission which would lead to longer network lifetime.

Multi-hop networks are not without drawbacks. For one, the messages take relatively longer to reach the destination due to the large number of hops. Moreover, the probability of a collision between packets increases as a consequence of multiple hops. In fact, the performance of both single-hop and multi-hop networks are affected by collisions between packets. The extent of damage caused by the collisions depends on the kind of scheme used for medium access. The *channel* is the medium through which data is transferred from source to the destination. Carrier Sense Multiple Access (CSMA) is one type of channel allocation scheme, in which the nodes in the network sense the channel prior to transmitting packets [12]. In CSMA, if the channel is sensed to be busy, the sensing node refrains from transmitting, avoiding a collision in the process. CSMA is described in more detail in Section 2.3. There are two main types of collisions prevalent in CSMA:

1. **Neighborhood collisions** Neighborhood collisions occur when packets from two nodes which are within transmission range of each other overlap either partially or completely in time. Since every node is connected to every other node in a single-hop network, it is obvious that

Figure 2.1: Hidden terminal scenario

all collisions in one-hop networks are *neighborhood collisions*.

2. **Hidden terminal collisions:** In multi-hop networks, an idle channel around the transmitter does not necessarily imply that the channel around the intended receiver is also idle. Consider the scenario in Figure 2.1. Node A upon sensing an idle channel at time t_1 , starts transmitting to B and would transmit for the packet duration T . Let us suppose, node C senses the channel at some time t_2 , such that $t_1 < t_2 < t_1 + T$. Since node C is not in the transmission range of A , node C would also sense an idle channel. Node C would thus start transmitting to B . This would result in a collision of the packets at B . This constitutes a *hidden terminal scenario*.

The next section outlines the goals of this work and the motivation behind pursuing them.

2.2 Goals and Motivation

Most of the work on ad-hoc wireless networks has focused 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 route 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. 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. 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 possibil-

ities 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 due to 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 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. 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 addition, unicast is not feasible in embedded sensor networks. Consider a scenario where millions of very simple sensors are randomly embedded in smart structure, e.g., a bridge, to sense and relay information to the controller at a remote location. It is hard to imagine these sensors communicating

through the air using antennas, etc. Instead, they are likely to use a physical medium as an embedded metal plate or wire mesh, or even the material constituting the body of the structure. Obviously, this is broadcast albeit possibly in a non-euclidean environment. Similarly, sensors could be infused in human bloodstream or could just be embedded on a person's clothing, like a smart coat. These sensors could monitor such things as heart rate, pulse, blood pressure, etc., and could communicate through the fabric or even through the body. Thus, in many cases, broadcast seems to be the only viable option for embedded sensor networks.

A major advantage of flooding or broadcast is the lack of a complex network layer protocol for routing, address and location management. This suits our notion of keeping the system simple and yet able to achieve global tasks through self-organization of the member nodes and their underlying communication mechanism. In this thesis, we propose an intelligent broadcast mechanism for Large Scale Sensor Networks(LSSN) utilizing the inherent parallelism of broadcast to achieve a more robust and scalable system. The protocol attempts to balance the redundancy of messaging paths needed for robustness with the need to control excessive messaging that causes collisions and waste of bandwidth (see Section 2.4 on Broadcast Storm Problem).

The next section gives an overview of the protocols used for channel access in traditional sensor networks.

2.3 Medium Access Protocols (MAC)

Packet-based computer communication networks are characterized by three major components: switches, channels and protocols. The switches (or nodes) are the hardware entities that house the data communication functions; the protocols are the set of rules and agreements among the communicating parties that dictate the behavior of the nodes, and the channel is the physical medium over which signals, representing data, travel from one node to another [20].

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. For a successful transmission, interference has to

be avoided or at the least controlled. Thus, the channel becomes a shared resource whose allocation is critical for network operation. The channel allocation schemes are commonly 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. Conflict-free transmissions can be achieved by allocating the channel to the users either statically or dynamically. The channel can be divided by providing the entire bandwidth to a single user for a fraction of the time (TDMA), providing a fraction of the frequency range to every user all the time (FDMA), or providing every user a portion of the code space as in spread-spectrum systems (CDMA). A block diagram representation of the classification is given in Figure 2.2.

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. Unlike conflict-free schemes, idle users do not consume any of the channel resources. In the following discussion, we describe two very simple and popular contention-based schemes.

Figure 2.2: Classification of Multiple Access Protocols

Aloha Protocols

Aloha [1] belongs to the class of contention-based protocols and has the distinction of being the first random access protocol. Random access protocols address the situation where the traffic is bursty. With bursty traffic, a user can be busy for short amounts of time and idle for the remaining stretches. When the channel assignment is fixed, as is the case with FDMA and TDMA, those idle periods amount to wasted bandwidth even though other users may have data to transmit. One solution would be to allow stations to transmit whenever they have data to send. The obvious result eventually would be collided data (i.e. two or more stations transmitting at the same time). Because there is no guarantee of successful transmission in advance, individual

users account for and recover from corrupted data packets.

Aloha has been used widely because of its simplicity and ease of implementation. *Pure Aloha* is the basic protocol for single hop systems with an infinite population generating packets according to a Poisson process. A newly generated packet is transmitted immediately hoping for no interference from other users. In a slight variation of the pure aloha protocol, the channel is divided into slots, the width of the slot being equal to the packet transmission time. In *slotted aloha*, users are restricted to start transmission of packets only at slot boundaries. When packets collide, they overlap completely instead of partially, thus reducing the wastage of the bandwidth. Slotted aloha is almost twice as efficient compared to pure aloha. The only additional requirement of the slotted system is that the users be synchronized with respect to a global clock.

Carrier Sense Multiple Access (CSMA) Protocols

In CSMA [12], 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*. Several different back-off strategies have been proposed in the literature [7, 5]. If the channel is

sensed to be idle, the node transmits its packet. In single hop networks using CSMA, a collision occurs only when two or more nodes sense the carrier within a propagation time of each other. It is evident that CSMA is an efficient medium access protocol for single hop networks.

In multi-hop networks, however, CSMA-based systems suffer from *hidden-terminal collisions* [23]. As a consequence of multi-hop, an idle channel around the transmitter does not necessarily imply that the channel around the intended receiver is also idle. This hidden terminal problem decreases the throughput that is achievable using CSMA in a multi-hop network. Rude-CSMA [17] gains back some of this lost throughput by transmitting sometimes even if the channel is sensed busy.

CSMA can be classified into *non-persistent* and *persistent* CSMA. In the non-persistent case, if a node senses the carrier busy after having generated a packet, the node backs-off for a random duration and senses the carrier again after the timeout expires. With nonpersistent CSMA, there could be situations where the channel is idle in spite of the nodes having packets to transmit because they might be in the back-off mode. This is avoided in the *persistent* scheme where a node, upon sensing a channel busy, persistently keeps sensing the carrier until it senses an idle channel and then transmits.

Besides the ability to sense a carrier, some local area networks have an additional feature, namely, the nodes can detect interference among several transmissions (including their own) while their transmission is in progress. This produces an interesting variation of CSMA, that is known as Carrier Sense Multiple Access with Collision Detection (CSMA/CD). The operation of all CSMA/CD protocols is identical to the operation of the corresponding CSMA protocols, except that if a collision is detected during a transmission, the transmission is aborted and the packet is scheduled for transmission at some later time.

In the following sections, we give a brief analysis of the performance of CSMA-based protocols for single-hop networks.

The vulnerable period in CSMA Networks

Assume a single-hop network with users aggregately generating packets according to a Poisson process with rate parameter λ . Let T be the time required for transmission of a packet and a the maximum propagation delay in the system.

Consider an idle channel and a node generating a packet at some time t (Figure 2.3). This node senses the channel, finds it free, and so starts transmitting at time t and does so for T seconds. The packet would take an addi-

Figure 2.3: Unsuccessful Transmission Period in Unslotted CSMA

tional a seconds to arrive at the destination. Thus a successful transmission causes the channel to be busy for $T+a$ seconds. If, at time $t+T+a > \hat{t} > t+a$ another node scheduled a packet for transmission, that node would sense the channel busy and hence refrain from transmission. But if some other node scheduled a packet for transmission during the period $[t, t+a]$, that node would sense the channel idle, transmit the packet, and cause a collision. Thus, the initial period of a seconds is called the *vulnerable period*. In case of a collision, the channel will be busy for some random duration between $T+a$ and $T+2a$ seconds [20].

Figure 2.4: Slotted Nonpersistent CSMA Packet Timing

The vulnerable period in slotted CSMA

In slotted CSMA, the channel is divided into mini-slots, the width of a mini-slot being equal to the maximum propagation delay. This scheme is illustrated in Figure 2.4, where the arrow marks represent the times the nodes in the network sense the channel. The shaded regions imply a collision between two or more packets while the unshaded regions signify a successful transmission. A collision occurs in slotted CSMA if two or more packets arrive within the same mini-slot. It is evident from Figure 2.4 that, in slotted CSMA, both successful and unsuccessful transmission periods last $T + a$ seconds. This is an improvement over the unslotted system in which an unsuccessful transmission period would consume anywhere between $T + a$ and $T + 2a$.

The success of a transmission in CSMA-based networks is not guaran-

teed, because CSMA does not employ any channel reservation techniques. The next section describes an efficient medium access scheme which allows the nodes to reserve the channel and thus eliminate most of the problems encountered in CSMA-based networks.

Multiple Access with Collision Avoidance (MACA)

As stated before, the CSMA scheme when used in the context of multi-hop networks frequently suffer from hidden and exposed terminal problems. This problem is overcome in Multiple Access with Collision Avoidance (MACA) [11]. In MACA, when a node wants to send data to another, it first sends a short Request-to-Send (RTS) packet to the destination. The receiver, if it is not busy, responds with a Clear-to-Send (CTS) packet. On receipt of the CTS, the sender sends its queued data packet(s). If the sender does not receive a CTS after a timeout, it resends its RTS and waits a little longer for a reply. This three-step process is called a *dialogue*. The node that sends the RTS and data packets is called the *initiator*, while the node that sends the CTS is the *responder*.

The key to collision avoidance is the effect that RTS and CTS packets have on the other nodes on the channel. When a node overhears an RTS addressed to another node, it inhibits its own transmitter long enough for

the addressed node to respond with a CTS. When a node overhears a CTS addressed to another node, it inhibits its own transmitter long enough for the other node to send its data. The initiator of the dialogue, includes in its RTS packet the amount of data it plans to send, and the responder echos that information in its CTS packet. The transmitter is inhibited for the proper time even if nothing is heard in response to an RTS or CTS packet.

As long as the link between each pair of nodes in the network is reciprocal (symmetric transmission radii), the receipt of a CTS packet by a node not party to a dialogue tells it that if it were to transmit, it would probably interfere with the reception of data by the responder (the sender of the CTS). MACA thus inhibits transmission when ordinary CSMA would permit it (and cause a collision), thus relieving the system of hidden terminal problems. Conversely, if a node hears no response to an overheard RTS, then it may assume that the intended recipient of the RTS is either down or out of range. In this case, MACA allows a transmission to proceed when ordinary CSMA would prevent it unnecessarily, thus relieving the system of exposed terminal problems. Even though collisions can occur between RTS packets, MACA still has the advantage over CSMA as long as the RTS packets are significantly smaller than the data packets. As long as this is true, collisions between RTS

packets are much less "costly" than the collisions that would otherwise occur between data packets. The savings in collision time also pays for the overhead of the RTS and CTS packets.

However, in broadcast-based networks, since the communication is one to all, the RTS/CTS exchanges cannot serve any useful purpose. The absence of MACA-like schemes for broadcast-based networks results in a variety of problems. The topic of discussion for the next section is one of the main issues encountered in broadcast-based systems.

2.4 The Broadcast Storm Problem(BSP)

Broadcast is widely used in mobile ad-hoc networks(MANET) for route discovery, paging a particular host and/or for sending an alarm signal. Because the nodes are geographically distributed, radio signals are bound to overlap. Thus, a straightforward broadcast by flooding is usually very costly and will result in serious redundancy, contention and collision. This is commonly referred to as the *broadcast storm* problem [18].

Broadcast Storm Problem Caused by Flooding

- **Redundant rebroadcasts:** When a node in a multihop network decides to rebroadcast a message, all/some of its neighbors might have

already heard the message from another neighbor.

- **Contention:** After a node broadcasts a message, if many of its neighbors decide to rebroadcast the message, these transmissions may severely contend severely with each other.
- **Collision:** Because of the deficiency of the back-off mechanism and the lack of RTS/CTS dialogue, collisions are more likely to occur.

Analysis of the Broadcast Storm Problem

The following analysis carried out in [18] makes an effort to quantify the cost of redundant rebroadcasts.

Figure 2.5: Additional area that benefits from a rebroadcast

Consider the scenario in Fig.2.1, where host A sends out a broadcast message, and it is received by node B among other nodes in the neighborhood. Let us say node B decides to rebroadcast the same message. Let S_A and S_B denote the areas covered by A's and B's transmissions, respectively. The

additional area that can benefit from B's rebroadcast is the shaded region, denoted by S_{B-A} . Let r be the transmission radius of A and B , and d the distance between A and B . Then, $|S_{B-A}| = |S_B| - |S_{A \cap B}| = \pi r^2 - I(d)$, where $I(d)$ is the intersection area of two circles with their centers at a distance d apart from each other.

$$I(d) = 4 \int_{d/2}^r \sqrt{r^2 - x^2} dx \quad (2.1)$$

The coverage area $|S_{B-A}|$ is maximum when $d = r$, and the maximum value equals $\pi r^2 - I(r) = r^2(\frac{\pi}{3} + \frac{\sqrt{3}}{2}) \approx 0.61\pi r^2$. This implies that a rebroadcast can only provide 0 to 61% additional coverage over that already covered by the previous transmission.

In reality, node B could be anywhere within A 's transmission range. The average value for coverage can thus be obtained by integrating equation 2.1 over the circle of radius x centered at A for x in $[0, r]$:

$$\int_0^r \frac{2\pi x[\pi r^2 - I(x)]}{\pi r^2} dx \approx 0.41\pi r^2$$

Therefore, a rebroadcast on an average can only give an additional coverage of 41%. A subsequent rebroadcast by some other node C after B 's transmission yields only 19% additional coverage. So rebroadcast of the

same message many times in the same neighborhood could do more harm than good for the success of that and other messages. The question that needs to be answered here is how beneficial would the rebroadcast of a message be after the node has heard the same message k times. The simulation results suggest that for $k > 4$ the *expected additional coverage* is below 0.05% [18].

Contention and Collisions

Carrier Sense Multiple Access (CSMA) is a widely accepted standard for medium access in broadcast based networks. Contention is a major cause for performance loss in CSMA-based systems. Let us suppose node A broadcasts a message and it is heard by n nodes. If the medium surrounding node A had been silent for a long time, then this broadcast from A would trigger a string of rebroadcasts from the n nodes that received the message. Some or all of these n nodes will be contending for the same wireless medium and this will result in certain collisions.

Consider the case when $n = 2$. Assume nodes B and C are the receiving nodes and B is located randomly in A 's transmission range. In order for C to contend with B , it must be located in the area $S_{A \cap B}$. Hence, the probability of contention is $|S_{A \cap B}|/\pi r^2$, assuming uniform density of nodes [18].

If x is the distance between A and B , then the expected probability of contention is

$$\int_0^r \frac{2\pi x I(x)/(\pi r^2)}{\pi r^2} dx \approx 59\%$$

Methods to counter the Broadcast Storm Problem

One sensible approach to the broadcast storm problem is to inhibit some nodes from rebroadcasting in order to reduce redundancy and, thus, contention and collision. Some of the existing methods presented in [18, 22] are described below.

Probabilistic Scheme: In this approach, each node on receiving a broadcast message will rebroadcast it with some fixed probability P . The problem with this scheme is that it does not consider the conditions prevailing in the system. The decision function is purely probabilistic and does not utilize any local information or intelligence.

Counter-Based Scheme: In this scheme, each node keeps track of the number of times, k , it has heard a message. Whenever k exceeds a chosen threshold, the node inhibits rebroadcast of the message. Though this method captures the traffic in the local neighborhood, a node on a critical path could decide against rebroadcasting.

Distance-Based Scheme: The relative distance between the nodes involved in a messaging process is used as the metric to make a rebroadcast decision. Let us suppose node X heard a broadcast message from node Y for the first time. If the distance, say d , between X and Y is very small, there is little additional coverage X 's rebroadcast can provide and there is no rebroadcast.

Location-Based Scheme: Nodes(equipped with GPS), use the absolute location information to evaluate the additional coverage their rebroadcast would provide and proceed only if this is significant.

Cluster-Based Scheme: This scheme assumes an underlying clustering algorithm that helps to form and maintain clusters. Nodes common to more than one cluster are called *gateway nodes*. On receiving a broadcast message, a non-gateway node — a node that belongs to only one cluster — refrains from rebroadcasting it further. A gateway node on receiving the broadcast, will use any of the above schemes to make a rebroadcast decision.

All the above schemes rely on various threshold mechanisms. Choosing a particular threshold involves a trade-off between reachability and ef-

efficiency. Several adaptive schemes have been proposed which dynamically adjust thresholds based on local connectivity information [22].

The following section gives an overview of broadcast-based multi-hop sensor network protocols.

2.5 Review of Data Handling in Sensor Networks

In this section, we list some of the most popular and widely accepted protocols for wireless sensor networks. The strengths and weaknesses of each of these protocols are outlined to give a better perspective of the challenges encountered in the design of sensor network protocols.

Flooding

Flooding is the most simplest case of broadcast where every message received by a non-destination node is flooded to all the node's neighbors. Flooding is extremely wasteful of bandwidth and creates a lot of collisions due to the simultaneous use of the wireless channel by multiple messages, all of which are lost as a consequence. In spite of these deficiencies, flooding is still useful because it is simple, relatively inexpensive and disseminates data quickly in a network where bandwidth is not scarce and links are not loss-

prone. Flooding is attractive from a complex system point of view because it exploits the inherent parallelism of the system to achieve robustness.

Gossiping

While flooding could be described as one-to-all communication, gossiping in communication networks is the process of distribution of the information known to each node to every other node in the network. In other words, gossiping is all-to-all communication. In gossiping, the sender node randomly picks one of its neighbor and forwards the message to this node. All nodes not addressed by the sender drop the message. Though gossiping suffers from all the problems of flooding, collisions are greatly reduced because the message gets only added to the queue of the node addressed by the sender. This results in less contention and fewer collisions. However, gossiping disseminates information at a slower rate compared to flooding. Another drawback of gossiping is that it is less robust to failures due to lack of redundancy. Also, the performance of gossiping deteriorates as the networks become larger due to cheap and, therefore, unreliable nodes that usually make up large networks.

Directed Diffusion

Directed diffusion [9] is data-centric in that all communication is for named data. Data is *named* using attribute-value pairs. A sensing task is disseminated throughout the sensor network as an *interest* for named data. This dissemination sets up *gradients* within the network designed to "draw" events (i.e., data matching the interest). Events start flowing towards the originators of interests along multiple paths. The sensor network reinforces one, or a small number of these paths.

Design Paradigm

- *Directed diffusion* is data centric.
- Data generated by sensor nodes is named by attribute-value pairs.
- A node requests data by sending interests for named data. Data matching the interest is then "drawn" down towards that node. Intermediate nodes can cache, or transform data, and may direct interests based on previously cached data.
- Interest and data propagation and aggregation are determined by localized interactions.

- An interest at a sink node is either flooded throughout the network in the absence of any information about potential interest satisfiers, or geographical routing is used to limit the topological scope of the interest diffusion resulting in energy savings.
- Data message is unicast individually to relevant neighbors.
- A sink node upon receiving data along multiple paths reinforces one of them for higher data rate.
- Design objectives include scalability, robustness and minimization of energy usage.

Desirable Features

- Potential for significant energy efficiency.
- Incorporates novel features like data-centric dissemination, reinforcement based adaptation and *in-network* data aggregation and caching for directional propagation.

Undesirable Features

- Caching requires storage space.
- Searching data and interest caches expends time and energy.

Adaptive Fidelity Algorithms

Adaptive fidelity algorithms [25] trade-off data delivery quality for battery lifetime, network bandwidth, or number of active sensors. Two application driven energy-conserving ad-hoc routing algorithms have been proposed. In the *Basic Energy-Conserving Algorithm (BECA)*, the idea is that nodes need not be listening and consuming power when they are not involved in transmitting, forwarding, or receiving data. Nodes in *BECA* transition between *sleeping*, *listening* and *active* modes. The node deployment density information is used to increase the time the radio is powered off in *Adaptive Fidelity Energy-Conserving Algorithm (AFECA)*. In densely-populated ad-hoc networks, many nodes are interchangeable for routing purposes. *AFECA* takes advantage of this information to increase *sleep times* of nodes when other equivalent nodes are available to forward data.

Design Paradigm

- Adaptive fidelity algorithms work above existing on-demand ad-hoc routing protocols, such as AODV [19] and DSR [10], without modification to the underlying routing protocols.
- Algorithms are based on the observation that, since networks consume

nearly as much power listening as receiving, the only way to substantially reduce energy consumption is by turning the radio off.

- Application-level information is used to turn-off the radios of the nodes to reduce energy consumption.
- The network lifetime is also improved by adaptively adjusting the routing fidelity based on the node density.
- *BECA* minimizes energy consumption by keeping radios powered-off as much as possible, trading higher *latency* for reduced energy use.
- *AFECA* adapts the number of nodes participating in ad-hoc routing to keep a constant level of routing fidelity and thus reduce energy consumption.

Desirable Features

- Complexity of the algorithm is independent of the size of the system.
- The type of application level information needed by the algorithms is very simple.
- In spite of turning off the radios, *BECA* establishes routes in all cases where a standard ad-hoc routing would.

- *BECA* and *AFECA* consume 50% less energy compared to that consumed by an unmodified ad-hoc routing protocol.
- Network lifetime scales well with node density.

Undesirable Features

- The cost of turning off the radio is added latency and more packet loss.
- Localized algorithms are usually parameter-sensitive.
- Simplified algorithms could be non-optimal under certain conditions.

Sensor Protocols for Information via Negotiation (SPIN)

The SPIN family of protocols [15] are used to disseminate information efficiently in a wireless sensor network. Conventional data dissemination approaches like flooding and gossiping waste valuable communication and energy resources sending redundant information throughout the network. In addition, these protocols are not resource-aware or resource-adaptive. SPIN solves these shortcomings of conventional approaches using data negotiation and resource-adaptive algorithms. Nodes running SPIN assign a high-level name to their data, called *meta-data*, and perform meta-data negotiations before any data is transmitted. This assures that there is no redundant data

sent throughout the network. In addition, SPIN has access to the current energy level of the node and adapts the protocol it is running based on how much energy is remaining. Simulation results show that SPIN is more energy-efficient than flooding or gossiping while distributing data at the same rate or faster than either of these protocols.

Design Paradigm

- Data in SPIN are named using high-level data descriptors, called *meta-data*.
- Negotiations of *meta-data* eliminate the propagation of redundant data through the network.
- Communication decisions are based on application-specific knowledge of the data and knowledge of available resources.
- Nodes in the network are constrained by energy, computation power and communication bandwidth.
- In SPIN, negotiation(that precedes data transmission) eliminates *implosion* (sending data to a node, which has already received the same from another source), use of meta-descriptors helps to eliminate *overlap*

(sensor nodes covering overlapping geographical areas), and nodes cut down their activity when energy is scarce, thus being *resource-aware*.

Desirable Features

- SPIN protocols deliver 60% more data for a given amount of energy than conventional approaches.
- Exchanging data about sensor data is less expensive than exchanging the actual sensor data itself.
- SPIN protocols closely match ideal dissemination protocols in both time and energy under certain conditions.

Undesirable Features

- There are costs associated with storage, retrieval and management of meta-data.
- The size of the meta-data descriptor should be smaller than the sensor data for SPIN to be beneficial.

Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [8] is designed for sensor networks where an end-user wants to remotely monitor the environment. In such a situation, the data from the

individual nodes must be sent to a central base station, often located far from the sensor network, through which the end-user can access the data. Some of the desired properties of protocols for these networks are use of uniform, battery-operated nodes, maximization of system lifetime and network coverage. Conventional network approaches, such as direct transmission, minimum transmission energy, multi-hop routing, and clustering all have drawbacks that don't allow them to achieve all the desirable properties. LEACH, which includes distributed cluster formation, local processing to reduce global communication and randomized rotation of the cluster-heads achieves the desired properties.

Design Paradigm

- LEACH is a self-organizing, adaptive clustering-based protocol.
- Localized coordination and control is used for cluster set-up and operation.
- Local cluster-heads are randomly rotated to distribute the energy load evenly among the sensors in the network.
- Scalability and robustness in dynamic networks is achieved through localized coordination.

- The routing protocol utilizes data fusion to avoid information overload.

Desirable Features

- A node's decision to be a cluster-head is made independent of other nodes in the network. Thus no extra negotiation is needed to determine the cluster-heads.
- LEACH achieves a factor of 8 reduction in energy dissipation compared with conventional routing protocols.
- LEACH results in random node death and hence does not leave any one subsection of the environment uncovered.
- LEACH improves the lifetime of the system considerably.

Undesirable Features

- The quality of the output is compromised by compressing the data.
- Data collection is centralized and periodic. A user might not need the data at all times.

Gradient Broadcast in Large Scale Sensor Networks (GRAB)

GRAB [4] explores a scenario where the sensor population is huge and unreliable, nodes have limited power, and topology continually changes due to

depleted and new sensors. The design has four major components. In the first component, a *cost field*, defined as the minimum cost from that node to the sink on the optimal path, is set up at each node. The *cost* is generic and can take the form of *hop count*, *consumed energy* or *delay*. Once the *cost field* is set up, data is delivered to the sink along the minimum cost path. The robustness of the design is further improved by controlling the *credit*, the total amount of energy given for a message minus the minimum required energy from source to sink, carried by the messages in the *credit-based mesh forwarding* component. This approach is based on the intuition that a message having an extra amount of credit over the minimum required energy can afford to travel more paths during the delivery process. These paths interleave and form a forwarding *mesh*. The more *credit* a message has, the wider the *mesh* will be. The width of the mesh is adjusted to achieve robust message delivery. A *density control algorithm* is also proposed, which maintains a subset of nodes in the working mode to ensure desired sensing coverage while allowing other redundant nodes to sleep most of the time and wake up occasionally. The last component performs the function of *refreshing* the cost field to enable availability of upto-date information at the nodes.

Design Paradigm

- A collaboration process among sensors in a local neighborhood is in place, to enable fusion of data, and thus filter out unwanted information.
- GRAB assumes that a fraction of the sensor nodes are equipped with GPS. Other sensors use a collaborative location estimation algorithm with the GPS-equipped nodes as reference points.
- The design has four major components, *Cost Field Setup*, *Credit-Based Adjustable Mesh Forwarding*, *Density Control* and *Cost Field Refreshing*.
- In the *cost field* component, a cost field starting from a given sink is set up. As the message travels from the source to the sink, each intermediate node decides to rebroadcast the message only if the consumed cost plus the cost from this node is equal to the source cost. Thus minimum cost path forwarding can be achieved without storing explicit path information.
- Each packet carries a cost credit in the *mesh forwarding* component. By assigning different credits, the width of the mesh and thus the

redundancy and robustness can be adjusted.

- Nodes are turned off to save energy. These nodes wake up at a later time to replace failing nodes. A constant *working density* of nodes is maintained to guarantee quality of forwarding and coverage.
- The cost field set up at the beginning might be rendered invalid due to topology changes. The sink node uses an event-driven or timeout approach to maintain an upto-date cost field gradient.

Desirable Features

- Information gradients enable self-organizing data flow.
- The protocol scales well with the number of sensors. In fact, the system lifetime increases in proportion with the number of deployed nodes.
- GRAB is robust to node failures and easily incorporates deployment of new sensors.
- Wasted effort due to mobile stimuli and mobile information sinks is minimized.
- GRAB incurs the overhead of setting up and maintaining cost gradients.

Undesirable Features

- Data fusion takes place at a Center of Stimulus(CoS) node which is fixed. Consequently, the CoS nodes have a higher rate of failure.

Rumor Routing in Sensor Networks

This approach proposes a novel method to route queries to a node that has observed an event [2]. An *event* is an abstraction, identifying anything from a set of sensor readings to the nodes processing capabilities. In this work, events are assumed to be a localized phenomenon, occurring in a fixed region of space. A *query* on the other hand, can be a request for information, or orders to collect more data. Once a query arrives at its destination, data flows backward to the query's originator.

Design Paradigm

- It is possible to flood either the query or the event itself. The former is used when the returning data from the sink is significant. When there are fewer events than queries, event flooding is preferred.
- The idea of *rumor routing* is to create paths leading to each event instead of a network-wide query. Thus, *rumor routing* fits in between query and event flooding.

- Each node generates an agent with some probability. An agent is a long-lived packet, which travels the network, propagating information about local events to distant nodes.
- A node, upon generating a query, transmits it if it has a route to the event. If it does not have a route, it forwards the query in a random direction.

Desirable Features

- Since the query is directly routed to the event as soon as the query discovers a path to the event, flooding is a rare occurrence.
- Below a certain threshold ratio of queries to events, rumor routing is more energy-efficient than event flooding.
- Achieves high delivery rate.

Undesirable Features

- The algorithm assumes no collisions.
- Performance is highly dependent on event distribution.

A summary comparison of the protocols described above is given in Table 2.1. Given the fact that all the protocols use a broadcast channel to flood either the event or the query, it is natural that they suffer from the *broadcast storm problem* discussed in Section 2.4.

Protocol	Energy-Aware	Scalability	BSP	Control Overhead
Flooding	No	High	Yes	No
Directed Diffusion	Yes	Medium	Yes	Yes
Adaptive Fidelity	Yes	High	Yes	No
LEACH	Yes	High	No	No
SPIN	Yes	Medium	No	Yes
GRAB	Yes	High	No	No
Rumor Routing	No	High	Yes	Yes

Table 2.1: Summary of the protocols discussed

In the next chapter, we describe the novel methods to facilitate *intelligent broadcast* proposed in this work.

Chapter 3

Methods

In this chapter, we first provide a brief description of the issues addressed in this thesis. We then outline the general approach adopted to overcome the problems. In the later sections, we describe the routing and MAC protocols proposed in this thesis.

3.1 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. In this thesis, we are more interested in how the communication process between geographically separated sensors self-organize rather than exploring issues related to user-sensor interactions, such as data querying and aggregation. 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. The sensors are then randomly distributed over this geographical space.
2. The direct communication range of individual sensors is much smaller than the size of the deployment region. In general, any pair of sensors is separated by a very 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. We

assume that the nodes do a Voronoi tessellation of the space and each node knows what region it is responsible for.

4. The sensors have minimal computational capability and very low energy. This means that there is no room for complex signal processing or optimization algorithms at the sensor level.
5. Sensors have relatively high failure rate, and are often temporarily out of commission due to energy limitations or physical factors. These failures should not affect performance and should be invisible to the user.

The following section, outlines the general approach adopted in this work.

3.2 Overall Approach

We start out with a system where all communication is broadcast. At the simplest level, this amounts to pure flooding, in which every message received by a non-destination node is flooded to the nodes neighbors. The only loop control in the system is based on the principle that no message will be re-broadcast more than once. 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 earlier, or rejected for re-broadcast earlier.

We then add simple heuristics to the above system, with efficiencies created by each node's decision whether to re-broadcast or not (depending on received data, its destination, and the local information available at the node). Finally, we study the possibility of using explicit feedback information to achieve superior performance. These schemes are discussed in more detail in the ensuing sections.

In addition, we also propose a novel scheme for medium access over which the above algorithms have been implemented. This is the topic of discussion for the next section.

3.3 An Improved Scheme for Medium Access

In a broadcast CSMA channel, the two major causes for performance loss are the *neighborhood* and *hidden-terminal* collisions as discussed in Section 2.1. Which of the two dominate, is determined by the traffic prevailing in the network and to some extent by the topology. Even though slotted CSMA improves over the unslotted version by synchronizing the nodes and thus

reducing the length of the unsuccessful transmission periods (refer Section 2.3), 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, albeit among messages generated at different times. To illustrate this, assume that a node A sets a back-off to some time t_1 upon sensing the channel busy at time $\hat{t}$. Assume that nodes B, C, D contend for a slot at some time $t > \hat{t}$, sense it busy and back-off to times t_1, t_2 and t_3 respectively. At time t_1 , after the back-off of node B expires, node B could sense the carrier free and transmit, but node B 's packet would still collide with the packet from node A from an earlier back-off. This is possible, because nodes in a slotted system, back-off to integral multiples of the slot. Thus, each slot is subject to collision among messages, generated at various times in the past, that happen to chose the same back-off point. We call this the problem of *colliding backoffs*. This is the 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 upon finding it idle would transmit the packet. 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, by using this CSMA protocol as it is, a successful transmission would trigger a spate of collisions in a relatively calm neighborhood.

We propose a novel channel access scheme, in which each node has a fair chance to earn a slot. This scheme is described in the following section.

3.3.1 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, namely, *CSMA with mini-backoff (CSMA-mb)*. CSMA-mb is built over the slotted CSMA protocol and shares the same philosophy apropos to the maximum propagation delay in the system. That is, in a multi-hop network, since the communication is localized (between one-hop neighbors), the maximum propagation delay, a , between the nodes is not too large. In pure slotted CSMA, a collision

Figure 3.1: Data structure of a slot in CSMA-mb

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 chose the same back-off point or both. 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. In other words, the main 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 main features of CSMA-mb are listed 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 Figure 3.1.
- 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.
- 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.
- Each one of the nodes that contend at the beginning of a major slot would thus 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.

Figure 3.2: Channel Access in CSMA-mb

- 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 also set the same mini-backoff value. Thus, the performance of the system can be improved further by increasing the resolution of the *contention period*.

The new scheme is explained with an example scenario in Figure 3.2. Assume that neighbors A, B, C and D 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, A, B, C and D , 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, B would sense the channel free when its mini-backoff expires and would start transmission immediately afterwards. Nodes A , C and D 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 A, B, C and D would have collided.

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. In the next section, we provide an analytical model for our system and in the process try to quantify the degree of caution exhibited by CSMA-mb.

3.3.2 Analytical Model

Development of medium access protocols and their performance evaluation has received a lot of attention over the years. In this section, we provide a model for the contention in a neighborhood for CSMA-mb, in the presence of hidden and neighborhood collisions.

Assumptions

1. The nodes in the network are distributed as a 2D Poisson point process with density γ .

The average number of nodes within a circle of radius R is then

$$N = \gamma\pi R^2 \quad (3.1)$$

The probability that there are i nodes within a circle R is given by

$$P(i) = \frac{N^i}{i!} e^{-N} \quad (3.2)$$

2. Every node with a non-empty queue tries to transmit with probability one. This means that,

$$P(a \text{ node tries to transmit}) = P(a \text{ node has a non empty queue}) \equiv p_q$$

We begin by defining some notation. Consider the transmission between two nodes X and Y , as shown in Fig. 3.3. Let area $N(X)$ represent the hearing range of node X and area $N(Y)$ represent the hearing range of node Y . Thus, X and Y can hear each others transmission. Transmissions from the nodes in the area $B(r)$ (shaded region) can be heard by Y , but not by

Figure 3.3: X to Y communication

X , where r is the distance between X and Y . Nodes in the area $C(r)$ can hear the transmissions of both X and Y .

Let $r = \text{Distance between two nodes } X \text{ and } Y$

$C(r) = \text{The area of region } N(X) \cap N(Y)$

$B(r) = \text{The area of region } N(Y) - N(X) \text{ in the Figure 3.3.}$

Figure 3.4: Area of the Chord

The area of the region A in Figure 3.4 when $R = 1$ is

$$q(b) = \cos^{-1}(b) - b\sqrt{1 - b^2}$$

Therefore,

$$C(r) = 2R^2q\left(\frac{r}{2R}\right)$$

and

$$B(r) = \pi R^2 - 2R^2q\left(\frac{r}{2R}\right)$$

With the assumptions and notations in place, we proceed to evaluate the throughput of CSMA-mb in the context of multi-hop networks.

Throughput of Persistent Slotted CSMA with Mini-backoff

In a multi-hop network, the medium around a node is shared by all its neighboring nodes. The actions of the neighboring nodes depend on their neighboring nodes. Hence, the precise computation of throughput is difficult. Thus, we use an approximate model similar to the one used in [12] to evaluate the success probability.

When X is ready to send a packet to Y , it senses the channel prior to its transmission. It is unable to sense the behavior of the nodes in $B(r)$, and the nodes in $B(r)$ become its hidden terminals. When Y is receiving a transmission from X , interference may occur from the nodes either in $C(r)$ or in area $B(r)$. The effect of each one of these interferences on the eventual success of a transmission from X to Y are analyzed in the following sections.

Neighborhood Collisions

In slotted CSMA, X can sense the transmissions from nodes in $N(X)$ in one slot and it starts a transmission only when it senses the channel to be idle. Hence, the condition that nodes in $C(r)$ do not interfere with transmissions from X to Y is that the nodes in $C(r)$ do not transmit in the same slot as X .

However, in CSMA-mb, the nodes set a mini-backoff upon sensing an idle channel for the first time in a major slot. Hence, the contentions are more spread out in time and as a result, the chances of a collision between two nodes are greatly reduced. Below, we derive a formal expression for this probability.

The probability that there are i nodes in $C(r)$ is

$$P(i) = \frac{(\gamma C(r))^i}{i!} e^{-\gamma C(r)}$$

If all mini-backoffs are selected with equal probability, the probability that a node in $C(r)$ will set the same mini-backoff as X is,

$$\begin{aligned} p_m &= \text{Prob}(\text{The node is ready to transmit}) \\ &\quad \times \text{Prob}(\text{The node selects the same mini-backoff value}) \\ p_m &= p_q \cdot \frac{1}{m} \end{aligned}$$

where m is the number of slots reserved for contention.

Therefore, the probability that no node in $C(r)$ sets the same mini-backoff as X is,

$$\begin{aligned}
p_c &= \sum_{i=0}^{\infty} (1 - p_m)^i \frac{(\gamma C(r))^i}{i!} e^{-\gamma C(r)} \\
&= e^{-p_m \gamma C(r)} \sum_{i=0}^{\infty} \frac{((1 - p_m) \gamma C(r))^i}{i!} e^{-(1-p_m) \gamma C(r)} \\
&= e^{-p_m \gamma C(r)}
\end{aligned}$$

As discussed in Section 3.3.1, CSMA-mb provides the nodes with a further opportunity to resolve contentions in a local neighborhood. This adds an additional probability, p_m — the probability that a node in $N(X)$ with a queued message would set the same mini-backoff as X — to the probability of collision in a neighborhood. As a consequence, the probability of a neighborhood collision, $1 - p_c$, in CSMA-mb is much smaller than the corresponding probability in pure slotted CSMA. This is one of the main reasons for superior performance of CSMA-mb. Next, we analyze the effect of hidden-terminals on performance.

Hidden-Terminal Collisions

Figure 3.5: The vulnerable period for CSMA protocols in multi-hop networks

The hidden-terminal problem has been described at length in Section 2.1. In Figure 3.3, the nodes in $B(r)$ constitute a hidden-terminal scenario for a transmission from X to Y . In slotted CSMA, the interference to a transmission from X to Y from nodes in $B(r)$ can occur at any time. To prevent this interference, it is necessary that nodes in $B(r)$ remain silent through the period beginning $\tau = \frac{T}{a}$ slots before X begins its transmission and ending one slot after X completes its transmission, where T is the length of the packet and a is the maximum propagation delay between two nodes in the system. This period is illustrated in Figure 3.5, and it is called the *vulnerable period* (VP). Hence, the condition that nodes in $B(r)$ do not interfere with transmissions from X to Y is that the nodes in $B(r)$ remain silent for $2\tau + 1$ slots of VP.

However, a hidden terminal collision will occur in CSMA-mb only if the

Figure 3.6: Analysis of hidden terminal collisions in CSMA-mb

hidden terminals in $B(r)$ sense the channel at the same time as X . If none of the hidden terminals are ready to transmit at the beginning of the major slot, the transmission from X will not be affected by hidden terminals for the entire length of its packet transmission. The next time the nodes in the system can sense the channel is at the beginning of the next major slot, by which time X would have completed its transmission. This condition is illustrated in Figure 3.6, where the width of the mini-slot is equal to the maximum propagation delay, a . If the length of the major slot is T , then $\tau = \frac{T}{a}$ is the number of mini-slots in a major slot.

Hence, the probability of no hidden terminal collision at Y is equal to the probability that none of the nodes in $B(r)$ transmit in the slot as X , which is

$$p_b = \sum_{i=0}^{\infty} (1 - p_q)^i \frac{(\gamma B(r))^i}{i!} e^{-\gamma B(r)}$$

$$\begin{aligned}
&= e^{-p_q \gamma B(r)} \sum_{i=0}^{\infty} \frac{((1-p_q) \gamma B(r))^i}{i!} e^{-(1-p_q) \gamma B(r)} \\
&= e^{-p_q \gamma B(r)} \\
&= e^{-p_q \gamma (\pi R^2 - 2R^2 q(\frac{r}{2R}))}
\end{aligned}$$

The nodes in CSMA-mb sense the channel only at integral multiples of a major slot and thus have fewer opportunities to interfere with a transmission. Thus, hidden-terminal collisions are reduced in CSMA-mb. This has another side to it: If none of the nodes have packets to transmit at the beginning of a major slot, then the channel will be idle for the entire duration of the major slot even if a node generates a packet midway through the major slot. This idle period can be reduced if the nodes sense the channel at integral multiples of the contention period instead of the major slot.

Altogether, the condition for a successful transmission from X to Y is that the nodes in $C(r)$ do not set the same mini-backoff as X and the nodes in $B(r)$ do not transmit in the same slot as X . Therefore, the probability of a successful transmission from X to Y is,

$$\begin{aligned}
Prob[X \longrightarrow Y | d(X, Y) = r] &= Prob(X \text{ transmits in a slot}) \\
&\times Prob(Y \text{ does not transmit in the slot}) \\
&\times Prob(\text{No transmission from } B \text{ during the slot}) \\
&\times Prob(\text{No ready terminal in } C \text{ sets the} \\
&\quad \text{same mini-backoff as } X) \\
&\times Prob(X \text{ sets the least back-off value})
\end{aligned}$$

$$Prob[X \longrightarrow Y | d(X, Y) = r] = p_q(1 - p_q)e^{-p_q\gamma(\pi R^2 - 2R^2q(\frac{r}{2R}))}e^{-p_m\gamma C(r)}p_x$$

where p_x is the probability that node X sets the least back-off of all nodes it is contending with.

Evaluation of p_x

Let L_b be the event that node X sets the lowest back-off value in its neighborhood, and N_X , the number of neighbors of X with a message, then,

$$\begin{aligned}
P(L_b | N_X = n) &= Prob(X \text{ backs-off to the first slot and rest of the} \\
&\quad n - 1 \text{ nodes to the following } m - 1 \text{ slots})
\end{aligned}$$

$+ \text{Prob}(X \text{ backs-off to the second slot and rest of the}$
 $n - 1 \text{ nodes to the following } m - 2 \text{ slots}) + \dots$
 $+ \text{Prob}(X \text{ backs-off to the } (n - 1)^{\text{th}} \text{ slot and rest of the}$
 $n - 1 \text{ nodes to the following one slot})$

$$\begin{aligned}
P(L_b | N_X = n) &= \frac{1}{m} \left[\frac{m-1}{m} \right]^{n-1} + \frac{1}{m} \left[\frac{m-2}{m} \right]^{n-1} + \dots + \frac{1}{m} \left[\frac{m-(m-1)}{m} \right]^{n-1} \\
&= \frac{1}{m^n} \sum_{i=1}^{m-1} (m-i)^{n-1} \\
&= \frac{1}{m^n} \sum_{i=1}^{m-1} i^{n-1} \\
&= \frac{1}{m} \sum_{i=1}^{m-1} \left(\frac{i}{m} \right)^{n-1}
\end{aligned}$$

Hence, the probability that X sets the least back-off value is,

$$p_x = P(L_b) = \sum_{n=1}^{\infty} P(L_b, n) = \sum_{n=1}^{\infty} P(L_b | N_X = n) P(n)$$

Assuming each node has a message with probability p_q independently of other nodes in the system, the average number of nodes in a neighborhood with message is Np_q , where N is the average number of nodes within a circle of radius R .

Therefore, the probability that there are n nodes with messages in the neighborhood of X is,

$$P(n) = \frac{(Np_q)^n}{n!} e^{-Np_q}$$

Hence, the final probability that a transmission from X to Y is successful, given X has a message is

$$Prob[X \longrightarrow Y | d(X, Y) = r, X_q] = (1 - p_q) e^{-p_q \gamma (\pi R^2 - 2R^2 q(\frac{r}{2R}))} e^{-p_m \gamma C(r)} p_x$$

where X_q is the event that X has a message.

Manfred Eigen, a famous mathematician, once said, “A theory has only the alternative of being right or wrong. A model has a third possibility: it might be right but irrelevant”. To dispel this third possibility, we ran simulations to verify the utility of our model, the results for which are provided in Chapter 4.

3.4 Intelligent Broadcast Scheme

Unconstrained flooding in ad-hoc sensor networks is not only wasteful of resources (bandwidth) but also inefficient in terms of reliable message delivery. At the same time the limitations of opting for sophisticated unicast-based algorithms for LSSN's are clear. It is thus essential to explore intelligent ways of using broadcast to achieve a robust and reliable system which takes advantage of the inherent parallelism and simplicity offered by such a scheme.

In this thesis, we propose an *intelligent broadcast (IB)* scheme, namely, *Self-organized Contention and Routing in Intelligent Broadcast Environments (SCRIBE)*, which works around the broadcast storm problem by adaptively controlling the redundancy of paths to obtain better performance. This scheme, in addition to reducing contention and consequent collisions, preserves the parallelism offered by broadcast-based schemes.

To this end, we explore an acknowledgement scheme for CSMA-mb, and use this feedback information to intelligently dequeue messages from the nodes. The motivation for this scheme is a similar approach used for efficient re-transmission with unicast-traffic to improve the performance of Time Spread Multiple Access (TSMA) protocols [14]. TSMA is a transmission scheduling scheme for a time-slotted system.

Motivation for Acknowledgements (Acks)

In large scale sensor networks, nodes depend heavily on local information to make decisions having global implications. In many situations, the information available at a node might just not be enough to make an intelligent decision. For example, a node's decision against rebroadcasting a message based on the number of times it has heard the message, could prove to be disastrous and even cause the message to be lost if the node happens to be on a critical path. In a similar vein, let us suppose that in a particular system, a node's re-broadcast decision function is governed by the node's perception of its neighborhood, i.e., a node in a dense neighborhood will rebroadcast with less probability compared to a node located in a sparse region. Assume that node A belongs to a dense neighborhood. There could be situations when node A decides not to rebroadcast a message (M_1), assuming that some of its neighbors would have received M_1 and thus would help in forwarding it. But since the nodes constituting an LSSN are highly unreliable, it is quite possible that many of node A 's neighbors were out of commission (due to shortage of energy, wear and tear, etc.) when node A received the message M_1 . It could also be the case that some of node A 's neighbors reacted to the message in the same way as node A did. Thus node A 's decision is not

justified.

In view of these considerations, we argue that the decision function of a node should be governed by *reliable* information rather than *available* information, i.e., in order to make a rebroadcast decision, a node should be able to gauge the actual importance of its rebroadcast. This motivated us to explore a scheme in which a node's decision to rebroadcast a message is based on reliable information regarding the successful reception of the message by any of the node's better placed neighbors (for that message). In other words, a node's decision to rebroadcast is influenced by the actual reception, or lack of it, of an explicit message, called *acknowledgement* (*Ack*), from any of the node's better placed neighbors for that message.

Ack for Broadcast Traffic

In a system with unicast traffic, if the destination node can instantaneously acknowledge the receipt of a packet to the source node, then the source node will have reliable information about the success of the packet and the need for retransmission. This could be realized by extending each data slot by including a fixed number of mini-slots, s , at the end of the data slot. The destination will use the s mini-slots, succeeding the data slot, to send an acknowledgement to the source node, if it successfully received a unicast

packet in the slots preceding the s mini slots. This scheme is based on the intuition that a successful transmission from a node A to another node B implies that none of node A 's neighbors could have successfully received a different message at the same time, so, B 's *acknowledgement* to A will also be received successfully.

To extend this scheme to work in a broadcast environment, the source node has to *select* a particular neighbor to acknowledge the reception of a data message. This node could be chosen at *random*, or even better, this could be the *most forward neighbor* with respect to the destination. Since our aim is to use this feedback information to dequeue the messages, we would like a more robust acknowledgement mechanism. Thus, instead of selecting one particular neighbor, a node selects k of its best neighbors for acknowledging the data message. But if all or some of the k neighbors of the sender node receive the message successfully, the Ack from these nodes would collide at the sender. Besides, we are more interested in the Ack from a receiver node successfully reaching the set of overlapping neighbors (between the sender and the receiver) that received the data message successfully in the first place. Therefore, to avoid the Ack collisions at the sender and intermediate nodes, the sender node *rates* the k neighbors, on a scale from 1 to k , according to

their geographical proximity to the destination of the message. This imparts a notion of relation quality among the neighboring nodes with respect to each message. The sender node encodes this information in the data message it transmits. Nodes transmit the Ack in the strict order of their rating. That is, a node with a lower rating acknowledges the data message only if it does not hear an Ack from a higher rated neighbor. Unrated nodes in the neighborhood with a data message, would drop the message from their queues upon receiving an Ack for that message from a rated neighbor. This approach allows the system to preserve the *best available redundancy* for each message. To set up the rating each node needs 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. The structure and functionality of control messages is explained in more detail in Chapter 4.

Ack collision scenario

Collision of acknowledgement packets though very rare, is still possible under the scenario depicted in Figure 3.7. Assume that node X transmits a broadcast message destined for node Y and is received by the neighbors of node X . Nodes A and B are better positioned than X to route the message

to Y .

Figure 3.7: Collision of acknowledgements

Assuming $k = 2$, node X selects nodes A and B to Ack the message. Let us suppose A is closer to Y than B . Node A gets a better rating than node B due to its geographical proximity to Y . Since, nodes A and B are out of range from each other, B will not hear A 's acknowledgement of X 's message. As a result, B would also start transmitting the Ack resulting in a collision of the acknowledgements. This collision is made possible due to the finite length of the acknowledgement packets.

Intelligent Dequeue Algorithm

Traditionally, Acks have been used as a basis for retransmission. The interesting aspect of SCRIBE is that we use Acks to obviate a scenario where a retransmission might be necessary. In other words, instead of waiting for a

collision to happen, and then retransmitting the packets, we try to reduce the possibility of a collision by dequeuing the unwanted redundant messages that might contribute to a collision.

The following is an algorithmic account of our dequeue mechanism;

Figure 3.8: Major Slot with Mini-slots for Ack in CSMA-mb

- Each data slot is extended by including s mini-slots at the end of the data slot. These s mini-slots are used for Ack contention and transmission. This scheme is illustrated in Figure 3.8.
- Owing to the fact that multiple neighbor nodes would receive a broadcast message from a source, the transmitting node selects k of its best neighbors to *Ack* the message and rates them according to their geographical proximity to the destination.
- The *neighbor selection process* is conditional on the fact that the neighbor node is at a better geographical position than the transmitting

node to forward the message to the eventual destination. If there are u nodes which are better placed than the source to route the message, then the transmitting node selects the $\min(u, k)$ neighbors and encodes this information in the message it sends out.

- Upon receiving a broadcast message, a node checks the Ack part of the message to see if it is one of the rated neighbors. If true, the node sets a mini-backoff proportional to its rating.
- This mini-backoff, is in principle, similar to the one used by the nodes to resolve contention for transmitting data. In this case, the contention being resolved is for Ack transmission. A rated node, upon the expiration of its mini-backoff, senses the channel and transmits the Ack in the event of an idle channel.
- There are scenarios under which an higher rated neighbor might not receive a message successfully. This could be the result of a collision at the node, or because the node had been out of commission during that major slot. In such an eventuality, a node with the next best rating would transmit the Ack. In general, a node with a lower rating transmits an Ack only if it fails to hear an Ack from an higher rated

neighbor. If it hears an Ack in the previous mini-slot, it inhibits its Ack transmission.

- A collision at the source node can only occur when two or more neighbors of the node transmit the Ack. This is true when two or more rated neighbors constitute a *hidden terminal scenario*. This situation is rare, because, the rated neighbors are closer to the destination than the source and hence will almost always be neighbors themselves.
- All nodes that receive the Ack but are not rated for that message, will drop the message from their queues, if present. This is an intelligent dequeue mechanism because, the reception of an Ack by a non-rated neighbor signifies that better placed nodes have received the message successfully and will route the message forward. In this way undesirable redundancy is reduced in the system.
- In addition, the lack of an Ack at the source could be used as a feedback to re-transmit the message and thus improve reliability.

In conclusion, *SCRIBE* exploits the *best available redundancy* in the system to achieve superior performance. But, it could be argued, like in any large system, that the true worth of a system lies in its resilience under failures. In the next chapter, we provide extensive simulation results of our system when operated under ideal and fallible conditions. We also compare our protocol with conventional protocols to ascertain the robustness of our scheme.

Chapter 4

Results and Discussion

In this chapter, we first present a brief description of the simulator and the metrics employed to evaluate the performance of the protocols. Our goals in conducting the evaluation study were two-fold: First, to place the performance of intelligent broadcast in the context of standard schemes such as *flooding* and *unicast*. Second, to understand the impact of network dynamics, such as node failures, on intelligent broadcast and to study the robustness of our protocol with respect to the idealized schemes.

4.1 Simulator Description

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 determine the connectivity of the network, given the linear size

of the network and the number of nodes, for various transmission radii. We then choose the R which just about guarantees full connectivity. For a 100 node network of linear size 50, full connectivity is achieved at a transmission radius of 8, whereas for a 500 node network of same size a transmission radius of 3 is required to ensure full connectivity. Messages are generated according to a Poisson process with mean rate λ . The message generation rate is chosen such that the network operates in a region far from overload. For our system, $\lambda = 0.05$ to 0.24 was found to be a reasonable range, given the simulation time and the size of a major slot. Some important simulation parameters are listed in Tables 4.1 and 4.2 below.

Parameter	Values
System Size	50×50
Number of Nodes	100/500
Radius	8/3
Generation Rate (λ)	0.05-0.24
Simulation Time	100000

Table 4.1: Network Parameters

The following section gives an account of the important steps involved in the simulation process.

Parameter	Values
Major Slot Size	85
Data Contention Period	30
Data Message Length	35
Ack Contention Period	6
Ack Message Length	14

Table 4.2: Channel Parameters

Simulator: Step by Step

The sequential steps involved in the network setup and operation are described below:

1. First, the nodes are randomly distributed with a uniform density over a flat square $2D$ surface of linear size l .
2. To start with, a node is only aware of its location, (x, y) , in the geographical space. During an initial set-up period, nodes exchange *control* or *hello* messages. The structure of a typical control message is $\langle \textit{node location}, \textit{transmission radius}, \textit{queue length} \rangle$. These are all attributes of the node. Through the exchange of these dedicated messages, a node determines the configuration of its neighborhood. Control messages are exchanged at a higher rate during the initial set-up process to facilitate *neighbor discovery*. These messages are still

exchanged for updating purposes, albeit at a lower rate, after data messaging starts in the system. In fact, when the traffic in the system is heavy, control messages could be piggy-backed to data, thereby, not creating additional traffic in the system. Also, during this period, the nodes perform a voronoi tessellation of their space and each node finds out the region it is responsible for. The sender of a message, addresses the message only by location, i.e., the sender is not aware if a node is present at that precise location, and the contention, as to who should eventually receive the message, is resolved at the destination.

3. After the set-up period, data messages are generated according to a Poisson process with mean rate λ . The source and destination for a message are chosen at random at generation time. The typical structure of a data message is $\langle \textit{message id}, \textit{source address}, \textit{destination address}, \textit{immediate source address}, \textit{hopcount}, \textit{hoplimit}, \textit{origin time stamp}, \textit{data} \rangle$.

The *message id* is the unique identifier assigned to a message at generation time. The *source* and *destination* addresses are the (x, y) co-ordinates of the random source and destination nodes of a message chosen at generation time. The *immediate source address* signifies the (x, y) co-ordinates of the one-hop neighbor from which the node re-

ceived a message for forwarding. While the *hop count* gives the number of hops a message has travelled thus far, the *hop limit* represents the maximum number of allowable hops for that message. The *origin time stamp*, as the name implies, is the time at which the message was generated. And finally, the *data* field represents the actual data being exchanged by the nodes.

4. The nodes in the system are synchronized with respect to a global clock. Thus, the nodes with non-empty queues contend for the channel at major slot boundaries. The node that wins the slot in a neighborhood, by setting the lowest mini-backoff among its neighbors, transmits the packet. If more than one node sets the same mini-backoff, the result is a collision of the packet at common neighbors.

Given below are the description of the routing protocols implemented for performance study.

Communication Protocols

To evaluate the robustness of the intelligent broadcast protocol, we compared our scheme with *pure flooding*, *pseudo-unicast* and *super-unicast* protocols described below. All the protocols described below were implemented using

a discrete-event simulator written in C.

1. In the *pure flooding* algorithm, a non-destination node upon receiving a message, blindly floods the message to the node's neighbors.
2. In the *intelligent broadcast* scheme, the nodes use the feedback information obtained from their one-hop neighbors to make a re-broadcast decision as described in section 3.4.
3. 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 [21]. Nodes in pseudo-unicast, as such, do not employ any channel-reservation scheme [11], and thus collisions are prevalent. Though nodes communicate using a broadcast channel, only the node addressed by the source receives the message while the rest drop the message. We look at pseudo-unicast in the first place because, by comparing this protocol with the intelligent broadcast scheme, one can quantify the effect the redundancy has on the performance of the system. Moreover, pseudo-unicast provides an ideal platform for comparison when the systems are subjected to failures. A comparison between pseudo-unicast and intelligent broad-

cast would illustrate whether a system with a single point of failure is sustainable under highly unreliable conditions or if it is necessary to preserve a certain redundancy level to achieve robustness under failures.

4. *Super-unicast* assumes idealized conditions for communication. Hence, a transmission from A to a neighbor B is always successful, provided B is in commission. In short, collisions are ignored in super-unicast. The motivating factor for investigating super-unicast is to compare intelligent broadcast 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 which are free from the above said problems, e.g. MACA. Instead of implementing any of these sophisticated protocols, we implemented the best-case unicast and compared it with the intelligent broadcast scheme. Hence, by comparing our scheme with both pseudo-unicast and super-unicast, we pretty much cover the entire domain of unicast protocols.

To study the effect of network dynamics on performance we evaluated the protocols under the following test scenarios.

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 system run. We explore this scenario to analyze the best-case performance of the protocols. By this, we obtain the maximum achievable performance of the protocols for a given set of parameters.
2. *Random failures:* One of the main limitations of a sensor network that differentiates it from traditional wireless networks is that the sensor nodes usually rely on a built-in battery as the primary source for their energy needs. This requirement is imposed by the concept of smaller and simpler nodes. Furthermore, the sensor nodes could become inoperative due to wear and tear. The nodes could also take timeouts to replenish their energy (recharge their batteries), and thus be inactive for the duration of the timeout. To model these actual periods of failure is a nontrivial problem. In the simulations, we incorporate the accumulation of these effects by introducing *random failures*. For

this, the nodes in the system are divided into *stable* nodes and *normal* nodes. Stable nodes are idealized, 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.

The next section lists the performance metrics used for comparison of the above protocols.

Performance Metrics

We chose two metrics to analyze the performance of *intelligent broadcast* and to compare it to other schemes.

1. *Average Message Delivery*: Delivery is defined as the ratio of the actual messages successfully received by the intended final destination to the total messages generated, for a given system run.

2. *Average Wait Time:* Wait time at a node signifies the elapsed time between the reception or generation of a message at a non-destination node and its subsequent transmission. In short, this metric defines the average time a message waits in a node's queue before it is transmitted. It is thus an indirect measure of the speed with which messages travel in the system.

We now proceed to present the results obtained through extensive simulation of our network.

4.2 Slotted CSMA Versus CSMA-mb

In slotted CSMA, since the nodes transmit immediately after sensing a free channel, neighborhood collisions contribute generously towards performance loss. Furthermore, the queues at the nodes are either short or empty, owing to the carefree, incautious nature of the protocol. In brief, a message generated is sent through the network almost instantaneously, invariably resulting in a collision. Thus the delivery achievable through slotted CSMA is inferior.

When broadcast by flooding is used as the means for communication, there is a need to bring the rate of message delivery up to a certain useful level. This prompted us to explore heuristics at the MAC layer to minimize

collisions, resulting in CSMA-mb. CSMA-mb could be characterized as a cautious protocol, which trades-off *wait time* at the nodes for improved *delivery*. As a result, messages in CSMA-mb are delivered with increased latencies compared to slotted-CSMA for the same number of hops. In other words, since the messages are transmitted blindly upon sensing an idle channel, wait time at the nodes in slotted-CSMA are much lower compared to those for CSMA-mb. On the other hand, nodes using CSMA-mb try to make sure that the channel is really free before transmitting. Owing to the cautious nature of CSMA-mb, one can observe a marked improvement in message delivery for CSMA-mb in Figure 4.1. It is also apparent from figure 4.1 that, as the length of the contention period is increased, neighborhood collisions are minimized, resulting in a more robust delivery percentage. In conclusion, this scheme leaves us with a flexible framework to achieve a desirable trade-off between message delivery and latency.

A model for contention in a local neighborhood when nodes use CSMA-mb for channel access was developed. Figure 4.2 shows the performance of the analytical model with respect to the simulation. It is apparent from the figure that the model is quite useful.

Figure 4.1: Performance comparison of slotted-CSMA and CSMA-mb. In this figure, the delivery achieved by flooding with slotted CSMA and CSMA-mb (for two different contention periods) for 100 nodes are depicted.

4.3 Effect of intelligent dequeuing on performance of flooding

In broadcast-based networks, every time a message is transmitted, a copy of it is received by many of the node's neighbors. In particular, the queues keep building up with every successful transmission. The messages are dropped or lost in the system only when collisions occur. Due to the cautious nature of the MAC protocol, CSMA-mb, even these collisions are kept to a minimum creating extremely long wait times in the process. Also, redundant copies of the messages lead to more contention and hence more collisions. It thus becomes necessary to dequeue those messages from the nodes which serve no useful purpose but just add to the undesirable traffic in the system. To put it differently, some kind of destructive mechanism for the messages is necessary to keep the system within bounds. It became evident from preliminary simulations that dropping off the messages probabilistically or just based on local information available at a node is not sufficient for significant improvement in performance. On the whole, the nodes in the system need some kind of feedback information to make a decision leading to an intelligent dequeue. This intuition formed the basis of the intelligent broadcast scheme proposed in this thesis. For the purpose of this analysis, we ran a set of simulations

Figure 4.2: This figure illustrates the performance of the model for one-hop transmission success in the system. The model is compared with the data from simulation for a 100 node network with message generation rate, $\lambda = 0.2$.

for the intelligent broadcast scheme for various values of k , the number of forward neighbors selected by the immediate source node to acknowledge its message. The results are analyzed in the following section.

Analysis of the Results

Performance curves for various values of k , the number of forward neighbors selected by the immediate source node to acknowledge its message, in the case of *intelligent broadcast* under infinite energy conditions are shown in Figures 4.3 and 4.4. Clearly, the case $k = 0$ amounts to *pure flooding*. A marked difference in delivery and wait times could be observed when k goes from 0 (pure flooding) to 1 — intelligent broadcast with minimum guaranteed redundancy, i.e., $k = 1$, implies that only one node is selected by a source node to Ack its message. Upon receiving an Ack from this node, the rest of the nodes with the same message would drop it unconditionally. For $k \geq 1$, $k = 6$ gives the best delivery and wait time, though the difference is only marginal. In addition, $k = 6$, is in agreement with the optimum number of node degrees required to achieve maximum throughput as suggested by Kleinrock et. al. [13].

Figures 4.5 and 4.6 depict the set of performance curves when the system

is subjected to random failures with 50% of the nodes being stable and the probability of failure in a major slot, P_f , for the normal nodes equal to 0.5. The performance of the system for various percentages of stable nodes for a given generation rate and P_f are shown in Figures 4.7 and 4.8.

From the performance curves, it is clear that maintaining a redundancy level based on the success of a transmission leads to improved delivery and wait times. Even under failures, the intelligent broadcast scheme performs better than flooding. This strengthens our argument that redundancy alone is not sufficient to achieve robust delivery. The redundancy has to be coupled with some amount of intelligence to offer the right amount of parallelism needed in the system. The need is thus to preserve the optimum but best available redundancy level rather than retaining all the redundant paths.

Figure 4.3: This figure illustrates the effect of intelligently dequeuing the messages on message delivery of flooding under infinite energy conditions. The delivery fraction is plotted for various values of k , the number of forward neighbors selected by the source to Ack its message.

Figure 4.4: This figure illustrates the effect of intelligently dequeuing the messages on wait times of flooding under infinite energy conditions. The wait times are plotted for various values of k , the number of forward neighbors selected by the source to Ack its message.

Figure 4.5: Delivery fraction of intelligent broadcast subjected to random failures with 50% stable nodes and probability of failure in a major slot, $P_f = 0.5$.

Figure 4.6: Wait times for intelligent broadcast subjected to random failures with 50% stable nodes and probability of failure in a major slot, $P_f = 0.5$.

Figure 4.7: Delivery fraction of intelligent Broadcast subjected to random failures with mean generation rate, $\lambda = 0.1$ and probability of failure in a major slot, $P_f = 0.5$.

Figure 4.8: Wait times for intelligent Broadcast subjected to random failures with mean generation rate, $\lambda = 0.1$ and probability of failure in a major slot, $P_f = 0.5$.

4.4 Protocols Comparison - Infinite Energy Conditions

This section focusses on the performance of flooding, intelligent broadcast, pseudo-unicast and super-unicast assuming infinite energy for the nodes. Figures 4.9 and 4.10 show the results for delivery and wait times for a 100 node system whereas Figures 4.11 and 4.12 are the results under similar conditions for 500 nodes.

Analysis of the Results

As stated before, the intelligent broadcast scheme performs much better than flooding both in terms of delivery and wait time. This improvement could be attributed to the dequeue mechanism that is in place in the intelligent broadcast protocol. The pseudo-unicast improves over intelligent broadcast protocol in smaller systems, because, all transmissions in pseudo-unicast are one to one, i.e., a source node picks the *most forward node*, with respect to the destination, among its neighbors and specifically addresses the message to this node. Though the transmission is broadcast, all nodes not addressed by the source drop the message. Thus, these messages do not get added to the queues of the nodes, resulting in less contention and fewer collisions. In

short, the loss in redundancy is more than made up by the reduction in traffic in unicast-based protocols for small networks. Super-unicast is the pick of the protocols under infinite energy conditions for small networks, because the only way a message is lost is if no path is available between source and destination. However, it should be noted that super-unicast is expensive, ideal, and not really practical.

While unicast-based protocols are better for smaller networks because reduction in traffic more than makes up for loss of redundancy, intelligent broadcast performs better as networks become larger. This effect is illustrated in the results for a 500 node network. This is mainly due to the fact that in large systems, the message has to travel a large number of hops before reaching the destination. As a result, the chances of a message colliding increases. In addition, the single-point of failure prevalent in unicast-based protocols is more prominent in large networks due to the increased exposure of such points to collision. This clearly illustrates the importance of preserving the redundancy in the system. In other words, the cautious redundancy of intelligent broadcast makes it a better protocol for larger systems at the expense of increased wait times. Another consequence of larger systems is dense neighborhoods. This effect is predominant in flooding, because a suc-

cessful transmission leads to increased contentions in a local neighborhood.

This explains the reason for poor performance of flooding.

Results for 100 Nodes

Figure 4.9: Delivery for Flooding, Intelligent broadcast, pseudo-unicast and Super-unicast under infinite energy conditions.

Figure 4.10: Wait times for Flooding, Intelligent broadcast, pseudo-unicast and Super-unicast under infinite energy conditions.

Results For 500 Nodes

Figure 4.11: Delivery for Flooding, Intelligent broadcast, pseudo-unicast and Super-unicast under infinite energy conditions.

Figure 4.12: Wait times for Flooding, Intelligent broadcast, pseudo-unicast and Super-unicast under infinite energy conditions.

4.5 Protocols Comparison - Random Failures

For the purpose of analysis under varying degree of failures, the network was simulated over various message generation rates for $P_f = 0.2, 0.5$ and 0.8 , with 50% stable nodes in the system. To study the degree of robustness in the system, the network was simulated over various percentages of stable nodes for a given message generation rate ($\lambda = 0.1$ and 0.2) and $P_f = 0.5$.

Analysis of the Results

It is apparent from Figures 4.13 to 4.18 that, as the probability of failure, P_f , increases, the broadcast protocols with their inherent parallelism, perform much better than unicast-based protocols. The redundancy in messages help in achieving a more robust delivery. To put it in context, the nodes in such unreliable systems go inoperative temporarily. In the case of unicast, if the most forward node is inoperative during a slot, the message is lost. In the intelligent broadcast scheme, chances are that at least some of the k neighbors would be operational in a given slot. If not true, the message could still be received by some other neighbors, apart from the selected k , in which case there will be no dequeuing in the system and redundancy is preserved. In the event that at least one of k neighbors was operational, the redundancy is

still preserved but also optimized by the dequeue mechanism. Figures 4.23 and 4.24 suggest the number of retransmissions required for super-unicast to match its performance under infinite energy conditions. This is clearly an overhead the system can do without.

Hence, for networks subject to random failures, the intelligent broadcast scheme outperforms all the other protocols because it works around the failures by maintaining redundancy based on the success of a transmission. Although *pure flooding* manages a better delivery rate than unicast protocols for low generation rates, the cost in terms of wait times is very high. This is where, the *intelligent broadcast scheme* is useful and attractive: it 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, Figures 4.19 to 4.22 show that the unicast-based protocols perform very poorly when more than half of the nodes in the system are unreliable. However, as the degree of unreliability decreases, the unicast protocols surpass the performance of their broadcast counterparts.

Figures 4.25 to 4.28 give the results for the 500 node system subjected to failures. A dramatic difference in the delivery rate can be observed between

unicast and broadcast protocols in Figure 4.25. Most of the arguments we made for larger systems under infinite energy conditions hold for systems subject to failures. However, the remarkable drop in delivery rate for unicast protocols could be attributed to the single point of failure that is prevalent in such networks. As networks become larger, the message needs to travel a large number of hops to reach the destination. The probability of a message collision increases with every hop in the network. All that is required for a message to be lost is for at least one of the nodes in the message path to be inoperative during a transmission. In contrast, there is no single point of failure in broadcast-based protocols. In fact, the intelligent broadcast scheme always maintains a certain redundancy under failures. This is the reason for superior performance of our scheme when compared to the unicast-based protocols.

Results for 100 Nodes: Effect of message generation rate

Figure 4.13: Delivery: Percentage of Stable Nodes = 50%, $P_f = 0.2$.

Figure 4.14: Wait Times: Percentage of Stable Nodes = 50%, $P_f = 0.2$

Figure 4.15: Delivery: Percentage of Stable Nodes = 50%, $P_f = 0.5$.

Figure 4.16: Wait Times: Percentage of Stable Nodes = 50%, $P_f = 0.5$

Figure 4.17: Delivery: Percentage of Stable Nodes = 50%, $P_f = 0.8$.

Figure 4.18: Wait Times: Percentage of Stable Nodes = 50%, $P_f = 0.8$.

Results for 100 Nodes: Effect of stable node percentage

Figure 4.19: Delivery: Generation Rate = 0.1, $P_f = 0.5$.

Figure 4.20: Wait Times: Generation Rate = 0.1, $P_f = 0.5$.

Figure 4.21: Delivery: Generation Rate = 0.2, $P_f = 0.5$.

Figure 4.22: Wait Times: Generation Rate = 0.2, $P_f = 0.5$.

Figure 4.23: Delivery of super-Unicast under failures over number of retransmissions of the message. Percentage of Stable Nodes = 50%, $P_f = 0.5$.

Figure 4.24: Wait times of super-Unicast under failures over number of re-transmissions of the message. Percentage of Stable Nodes = 50%, $P_f = 0.5$.

Results for 500 Nodes: Effect of message generation rate

Figure 4.25: Delivery: Percentage of Stable Nodes = 50%, $P_f = 0.5$.

Figure 4.26: Wait Times: Percentage of Stable Nodes = 50%, $P_f = 0.5$

Results for 500 Nodes: Effect of stable node percentage

Figure 4.27: Delivery: Generation Rate = 0.1, $P_f = 0.5$.

Figure 4.28: Wait Times: Generation Rate = 0.1, $P_f = 0.5$.

4.6 Hop Limit heuristic

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 to millions of nodes, 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 distributiveness of the system to ensure communication.

The intelligence could be derived either from the local information available at a node or from the information carried by the data messages. In addition, nodes could 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 section, 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 known as the *hop limit*. The notion of hop limit has been explored by researchers in the past. In *dynamic source routing* [10], 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* [6], a predefined higher limit on the number of hops over which a message may propagate is assigned.

Our contention is that, in a LSSN, some messages might be destined 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 (measured in hops) that it needs to traverse.

The hop limit thus assigned is decremented at every hop a message makes in the system. A non-destination node, upon receiving a message for forwarding, rebroadcasts the message only if the hop limit has a non-zero finite value. The results showing the effect of this heuristic follow. Figures 4.29 to 4.32 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 has been justified. In fact, the performance obtained by employing this notion of hop limit 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 results are consistent for the 100 and 500 node networks.

Results for 100 nodes

Figure 4.29: Delivery for flooding with and without hop limit on messages

Figure 4.30: Wait times for flooding with and without hop limit on messages

Results for 500 nodes

Figure 4.31: Delivery for flooding with and without hop limit on messages

Figure 4.32: Wait times for flooding with and without hop limit on messages

4.7 Summary of the Results

In this chapter we demonstrated through simulation results that the intelligent broadcast protocol is a viable one for LSSN's. We also presented comparisons of our protocol with pure flooding and unicast-based protocols under ideal and fallible conditions and analyzed the results. Under infinite energy assumption, we observed that though the unicast-based protocols performed much better than their broadcast counterparts for smaller systems, the intelligent broadcast scheme was found to be superior to unicast protocols for larger networks. We suggested that for unicast-based smaller networks the loss in redundancy is more than made up by the reduction in traffic. However, the large number of hops and increased probability of collisions make preserving redundancy a necessity rather than a luxury in larger networks. This we believe, resulted in the improved performance of intelligent broadcast for larger systems under infinite energy conditions.

We also presented results for the system when subjected to random failures. We observed that broadcast with its inherent parallelism is much more robust to failures than unicast protocols with directed messaging. It was also apparent from the results that the effect of preserving redundancy is more

prominent in larger systems. This could be attributed to two reasons. First, the chances of a message collision are more in larger networks due to the large number of hops. Secondly, the single point of failure of unicast-based networks expose the larger systems with long paths formed by unreliable nodes to more failures, resulting in messages being lost, than smaller systems. Also, once a message is lost, the only way it can be recovered is through re-transmission of the message by the source or a intermediate node that has the message in its cache. This would require a robust acknowledgement mechanism, reliable exchange of Ack packets, which in turn would use up some of the precious bandwidth. In addition, the re-transmitted messages would cause extra contention resulting in more collisions. We thus conclude that the cautious redundancy of intelligent broadcast makes it a more viable protocol for large networks made up of cheap and, therefore, unreliable nodes.

All the while, we have just been looking at the positive aspects of our scheme. But there are some obvious drawbacks. For one, the speed with which the messages reach the destination is considerably reduced due to the cautious nature of the MAC protocol, CSMA-mb. However, the most important disadvantage is the overhead of mini-slots for data and Ack contention.

As a result, the size of a packet in the intelligent broadcast scheme is larger compared to that of pure flooding and unicast protocols. In smaller networks, this is clearly an overhead that we can do without, at least in systems made up of reliable nodes. However, as the system becomes larger, resulting in cheaper and, therefore, unreliable nodes, the overhead of mini-slots is almost non-existent. This is because, due to the increased rate of collisions and high rate of node failures in larger systems, a message directed to a specific node has to be re-transmitted repeatedly to achieve satisfactory performance. This effect is illustrated for super-unicast in Figure 4.23. This not only results in the extra overhead of dedicated acknowledgement packets, but also results in more collisions. This is clearly avoided in the intelligent broadcast scheme by maintaining a cautious redundancy based on the success of a transmission, and thus obviating the need for re-transmissions. Hence, it can be concluded that the cost of extra mini-slots is outweighed by the improved and robust delivery rate achieved by intelligent broadcast for larger systems.

In selection diversity forwarding (SDF) [16], a slightly different acknowledgement scheme is used to facilitate intelligent rebroadcast. We stumbled upon this work well after the formulation of the SCRIBE algorithm. In the next chapter we conclude our work and suggest avenues for future work.

Chapter 5

Summary and Conclusion

In this chapter we conclude our work and suggest avenues for future work in this area.

5.1 Summary

Large Scale Sensor Networks (LSSN) are formed by randomly deploying a very large number of networkable sensors over an extended environment such that the sensors organize themselves into a network. The individual sensors comprising the network are highly unreliable, unattended and relatively unimportant for proper functionality of the network. In addition, the sensors only have limited computational and communication capabilities. A LSSN, in our vision, will consist of many dynamically interacting entities that give rise to a complex adaptive system. This complex behavior is in fact desir-

able, since it allows the system to accommodate and adapt to unforeseeable situations.

As discussed in Chapter 4, traditional routing protocols turn out to be inadequate and inappropriate for LSSN's. The need is thus for the system to self-organize the communication between the sensor nodes to provide robust, adaptable and scalable performance. It is also important to preserve the inherent parallelism offered by the system to maximize fault tolerance. Hence, the most suitable approach from the point of view of robustness and scalability is broadcast.

Broadcast by "flooding" is highly inefficient and wasteful of resources. At the same time, the parallelism offered by flooding is useful and should be preserved in large systems. The trade-off is thus between preserving the available redundancy to achieve robustness and obviating undesirable redundancy to minimize contention and consequent collisions in order to attain superior performance. The challenge lies in adapting the redundancy intelligently and as needed. This places a premium on the local information that is required to arrive at such a decision. We have shown that an intelligent decision in this regard can be made by exchanging explicit feedback information, in the form of acknowledgements, between the nodes.

Unlike the purpose of traditional acknowledgement schemes, the key to *intelligent broadcast* is the effect the Ack packets have on other nodes in the channel. An analogy could be drawn between this scheme and MACA, which achieves superior performance due to the effect the RTS/CTS packets have on other nodes in the neighborhood. As long as the Ack packets are significantly smaller than data packets, intelligent broadcast is useful. When this is true, collision between Ack packets are less “costly” than collisions that would otherwise occur between data packets if the redundancy is left unchecked. The savings in collision time pays for the overhead of Ack packets. In summary, cost is outweighed by benefit in intelligent broadcast. A summary of the merits and demerits of our scheme is given below:

Advantages of SCRIBE

1. The self-organized contention process in the intelligent broadcast scheme leads to improved delivery when compared to that of flooding and unicast-based protocols.
2. The intelligent broadcast scheme is more robust to failures than conventional unicast-based protocols. The robustness is achieved by adapting to a certain redundancy level intelligently and as needed.

3. This scheme gives superior performance for large networks due to its ability to maintain a cautious redundancy.
4. Intelligent broadcast provides a robust and reliable acknowledgement mechanism for re-transmission.
5. The system is highly scalable and robust.

Disadvantages of SCRIBE

1. The cautious nature of CSMA-mb has a negative impact on the speed with which the messages reach the destination. In other words, improved delivery is achieved at the cost of increased wait times at the nodes.
2. This scheme imposes the extra overhead of mini-slots for data and Ack contention.
3. Since the nodes contend only at major slot boundaries, an idle channel at the beginning of a major slot would mean that the channel is not utilized for the entire length of the major slot.

5.2 Conclusion and Future Work

In this thesis, we first propose Carrier Sense Multiple Access with mini-backoff (CSMA-mb), which is a slight variation of the slotted CSMA protocol. CSMA-mb improves over slotted CSMA by self-organizing the contention process. The delivery rate achieved by CSMA-mb is better than that of slotted CSMA protocol. The overhead of CSMA-mb is the extra mini-slots for contention and increased message latency. Next, we present *intelligent broadcast*, which adapts the redundancy in the system based on the success of a message to achieve robust performance.

Many issues left unexplored by the present work, and these provide extensive scope for future work in this area. This area promises to be a very fertile one for research. Some suggestions for future work in this area:

- In our scheme, the sender node selects the neighbors to Ack its message. Instead, the process could be made more distributed by moving the decision making to the receiver of a message. Moreover, the number of neighbors chosen to Ack a message could be based on a node's perception of its neighborhood and the traffic prevailing in the system instead of a predefined maximum.

- In the present system, all rated neighbors queue the message while the non-rated neighbors drop it unconditionally upon receiving an Ack for a message. Instead, the rated neighbors could dequeue a message with some probability upon receiving an Ack from another rated neighbor. This probability could be weighted by the rating of the neighbor from which the Ack was received. Similarly, a non-rated neighbor could dequeue the message with a probability based on the acknowledging node's rating.
- We could imagine another scenario where the mini-slots are extended to accommodate multiple acknowledgements. The length of an Ack packet can be made equal to one mini slot. The rated neighbors upon receiving a message, could Ack one by one in the strict order of rating. A non-rated neighbor could thus make a rebroadcast decision based on the number of Acks it received. This is in contrast to the present scheme where, only the node with the highest rating is encouraged to Ack.
- Mobility is an issue we left unexplored in our work. Our scheme relies heavily on accurate information about a node's neighbors. Mobility

would thus warrant frequent exchange of control messages for up-to-date neighbor information. This in turn would result in more traffic in the system. Also, the mobility would affect the connectivity in the network. Thus, it would become necessary to maintain a connected network at all times to achieve superior performance. Hence, it would be interesting to study the performance of LSSN's under highly mobile conditions.

- To study the extent of fault tolerance in the system, we introduced random failures. A more accurate performance evaluation can be obtained by explicitly assigning finite energy to the nodes and by using more detailed failure modes.

Bibliography

- [1] Norman 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.*
- [2] David Braginsky Daveey. Rumor Routing Algorithm For Sensor Networks. *citeseer.nj.nec.com/462331.html*
- [3] Deborah Estrin, Ramesh Govindan, and John Heidemann. Scalable coordination in sensor networks. *Technical Report 99-692, University of Southern California, January 1999.*
- [4] Fan Ye. GRAdient Broadcast: A Robust, Long-lived Sensor Network. *<http://irl.cs.ucla.edu/papers/grab-tech-report.ps>*
- [5] L.A. Goldberg and P.D. MacKenzie. Analysis of practical backoff protocols for contention resolution with multiple servers. *Journal of Computer*

and Systems Sciences, 58:232–258, 1999.

- [6] 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*.
- [7] J. Hastad, T. Leighton, and B. Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM Journal of Computing*, October 1995.
- [8] 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*.
- [9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *In Mobile Computing and Networking, 2000*.
- [10] Johnson David B., David A. Maltz. Dynamic Source Routing in Ad-Hoc Wireless Networks, *Mobile Computing*, 153-181, 1996.

- [11] P. Karn. MACA - a new channel access method for packet radio. *ARRL/CRRL Amateur Radio 9th Computer Networking Conference 1990*.
- [12] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part-I - carrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transactions in Communications*, COM23 (12):1400–1416, 1975.
- [13] 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*.
- [14] R. Krishnan and James P.G. Sterbenz. An Evaluation of the TSMA Protocol as a Control Channel. *BBN Technical Memorandum No. 1279 Job No. 11982000 26 April 2000*
- [15] J. Kulik, W. R. Heinzelman, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *In Proc. 5th Intl. Conf. Mobile Computing and Networking, pages 174–185, 1999*.

- [16] Peter Larsson. Selection Diversity Forwarding in a Multihop Packet Radio Network with Fading Channel and Capture. *Ericsson Research, Air Interface Architectures*
- [17] R. Nelson. and L. Kleinrock. Rude-CSMA, A Multihop Channel Access Protocol. *IEEE Transactions on Communications*, Vol. COM-33, No. 8, pp. 785-791, August 1985.
- [18] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, Jang-Ping Sheu. The Broadcast Storm Problem in Mobile Ad-Hoc Network. *In Int'l Conf. on Mobile Computing and Networking(MOBICOM)*, Aug. 1999.
- [19] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc On-Demand Distance Vector Routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.
- [20] Raphael Rom, Moshe Sidi. Multiple Access Protocols; Performance and Analysis. *SpringerVerlag*, 1990.
- [21] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. on Comm.*, 32(3):246–257, March 1984.

- [22] Yu-Chee Tseng, Sze-Yao Ni, En-Yu 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.*
- [23] F. A. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II - the Hidden Terminal Problem in Carrier Sense Multiple-access Modes and the Busy-tone Solution. *IEEE Trans. on Comm., vol. 23, no. 12, pp. 1417-33, 1975.*
- [24] L. Wu and P. Varshney. Performance Analysis of CSMA and BTMA Protocols in Multihop Networks: Part I - Single Channel Case. *Information Sciences, 120(1-4): 159-177, 1999.*
- [25] Ya Xu, John Heidemann and Deborah Estrin. Adaptive Energy-Conserving Routing for Multihop Ad-Hoc Networks. *Research Report 527, USC/Information Sciences Institute, October 2000.*