Network Wide Broadcasting Protocols for Mobile Ad Hoc Networks

by

Bradley B. Williams

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematical and Computer Sciences).

Golden, Colorado

Date __________________

Signed: ______________________________

Bradley B. Williams

Approved: ______________________________

Dr. Tracy Camp
Associate Professor

Golden, Colorado

Date __________________

____

Dr. Graeme Fairweather
Professor and Head
Department of Mathematical and Computer Sciences

ABSTRACT

Network wide broadcasting in Mobile Ad Hoc Networks (MANETs) provides important control and route establishment functionality for a number of unicast and multicast protocols. Considering its wide use as a building block for other network layer protocols, the MANET community needs to standardize a single methodology that efficiently delivers a packet from one node to all other nodes. Despite a considerable number of proposed broadcasting schemes, no comprehensive comparative analysis has been previously done. This thesis provides a theory-based and simulation-based comparison of the different protocols. Broadcast protocols are first described and classified into four categories. A run-time analysis of algorithm complexity is performed and a theoretical model to predict the fraction of retransmitting nodes is provided for two protocols. In addition, we simulate a subset of the protocols from each category, thus supplying a condensed but comprehensive side by side comparison. The simulations are designed to pinpoint, in each category, specific failures to network conditions that are relevant to MANETs, e.g., bandwidth congestion and dynamic topologies. Protocol extensions using adaptive responses to network conditions are proposed, implemented and analyzed for one broadcasting scheme that performs well in the comparative study. Two protocols utilizing neighbor knowledge are shown to be most applicable for a wide range of network conditions. Areas of future research are defined to perfect these two protocols, with the hope that one will be chosen as a standard broadcasting scheme by the MANET community.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background

Mobile Ad Hoc Networks (MANETs) are wireless networks which are characterized by dynamic topologies and no fixed infrastructure. Each node in a MANET is a computer that acts as both a host and a router and, as such, may be required to forward packets between nodes which cannot directly communicate with one another.

MANETs, as self configurable networks, are appropriate for the following situations in which a fixed infrastructure may not be feasible:

- Disaster relief/rescue operations in which emergency personnel need a temporary and quick-to-install communication infrastructure at a remote site.
- Military field operations which are dynamic in nature and which benefit from a high level of tactical information exchange.
- Commercial operations that allow information to be passed among network participants based on their physical and temporal locality. Geographic based advertising, pervasive computing, and conference/meetings are examples.

MANET challenges are present at all layers of the OSI network model. At the physical layer, the wireless medium is subject to signal interference and a finite communication range proportional to transmitter power. Unlike Ethernet, the Link Layer of a wireless network is not able to detect collisions during transmission; therefore, reliability must be ensured via mechanisms such as direct acknowledgment of reception. The network (or routing) layer must adapt to a dynamic topology in which neighbor sets change, sometimes frequently, over time. In MANETs, a hierarchical routing structure, such as that of the Internet, is difficult and expensive to maintain when links change frequently. At the transport layer, protocols must be able to differentiate between congestion and frequent packet loss in order to deliver a reliable stream of data.

Recent research in network layer protocols for MANETs has involved unicast, multicast and broadcast communications. Unicast protocols allow one mobile node to communicate with another mobile node across the network. Protocols described in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] all provide means for unicast communication. Researchers have also studied the use of geographic information, obtained via a system such as the Global Positioning System (GPS), in unicast routing [12, 13, 14, 15, 16].

Multicast communication involves one source node sending data to a number of destination nodes. Recent work in multicast communication for MANETs include [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28].

Network wide broadcasting, simply referred to as "broadcasting" for the remainder of this thesis, is the process in which one node sends a packet to all other nodes in the network. Broadcasting may be used to disseminate data to all other nodes in the network *or* may be used by MANET unicast or multicast routing protocols to disseminate control information. For example, many unicast routing protocols such as Dynamic Source Routing (DSR) [2], Ad Hoc On Demand Distance Vector (AODV) [29], Zone Routing Protocol (ZRP) [30], and Location Aided Routing (LAR) [14] use broadcasting or a derivation of it to establish routes. Currently, these protocols all rely on a simplistic form of broadcasting called Flooding, in which each node (or all nodes in a localized area) retransmits each received unique packet exactly one time. The main problems with Flooding are that it typically causes unproductive and often harmful congestion at the physical layer, as well as inefficient use of node resources.

1.2 Description of Work

Recently, a number of research groups have proposed more efficient broadcasting techniques whose goal is to minimize the number of retransmissions while attempting to ensure that a broadcast packet is delivered to each node in the network. None of these efficient broadcast schemes have been adopted by the unicast protocols mentioned above, perhaps because a unified comparison of these protocols over a wide range of MANET conditions was lacking. This work addresses that deficiency.

Our approach is to first describe how each protocol operates. We categorize the protocols into four groups by their inherent characteristics. Next, we provide an analysis of the complexity of each protocol in terms of a Big-O analysis of its run-time. The run-time complexity is analyzed in terms of three essential functions: Duplicate Packet Detection, Neighbor Table Maintenance, and the Core Algorithm for Retransmission Determination. Since each protocol distributes computational load differently across these functions, the aggregate complexity allows us to compare the different protocols.

Mathematically predicting retransmission frequency of these protocols also provides a useful tool for protocol comparison and allows us to determine which network conditions most affect protocol performance. We develop a model to predict the fraction of retransmitting nodes for two protocols given the inputs of network area, number of nodes, and node transmission range.

These prediction models were only developed for protocols with relatively simple algorithms; other algorithms are too complex to easily model. In addition, the models do not account for other important MANET conditions such as constrained bandwidth and dynamic topologies. Thus, we rely on our simulation analysis to provide a more detailed comparison of the different broadcast protocols. We simulate sample protocols from each category to allow us to make broad inferences regarding a broadcast category's operation and applicability.

Our simulation comparisons are designed to test the protocols under specific conditions of increasing neighbor density, traffic rates and node mobility. Each simulation study allows us to pinpoint deficiencies in the protocols and propose solutions to correct for specific problems.

We classify and evaluate all broadcast protocols (of which we are aware) that are distributed in nature and designed for asynchronous Medium Access Control (MAC) schemes. Our simulations utilized the IEEE 802.11 MAC specification [31]. While a significant amount of work in both hierarchical broadcasting protocols and broadcast friendly MAC protocols has been done, we do not consider that body of work within this thesis.

The remainder of this thesis is organized as follows. In Chapter 2, Section 2.1 describes attributes that are common to many, or all, of the protocols described. Section 2.2 describes known broadcasting schemes and the categorization of them. We then briefly summarize the protocols according to their features in Section 2.3. Chapter 3 begins with a background discussion on theory-based analysis (Section 3.1), and then provides the run time complexity analysis of each protocol (Section 3.2) and the derivation of the models that predict retransmission frequency for two protocols (Section 3.3). Chapter 4 deals with the simulation comparison of five broadcast protocols. In Section 4.1 we explain which protocols were selected for simulation and why. Section 4.2 provides an outline of the studies we performed and the simulation parameters we chose. Section 4.3 then gives the results from the simulations. Lastly, Chapter 5 presents conclusions and future work.

Chapter 2

NETWORK WIDE BROADCASTING PROTOCOLS

2.1 Common Attributes

In this section, we describe relevant features of the 802.11 MAC because its operation affects all broadcast protocols evaluated in this thesis. We then present two common attributes which exist in many (or all) of the broadcast protocols categorized.

2.1.1 802.11 Medium Access Control Specification

This thesis evaluates broadcast protocols on wireless networks that utilize the IEEE 802.11 MAC [31]. This MAC follows a Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) scheme. Collision avoidance is inherently difficult in MANETs; one often cited difficulty is overcoming the hidden node problem, where a node is not able to ascertain whether its neighbors are busy receiving transmissions from an uncommon neighbor.

The 802.11 MAC utilizes a Request To Send (RTS) / Clear To Send (CTS) / Data / Acknowledgment procedure for unicast packets. Having specific neighbors positively verify that they are free to receive and then acknowledge a transmission corrects for collisions at the receiving node. The RTS/CTS/data/ACK procedure is too cumbersome to implement for broadcast packets as it would require responses from all neighbors, which is something difficult to coordinate and bandwidth expensive. Therefore, the only requirement made for broadcasting nodes is that they assess a clear channel before broadcasting. Since a sending node only verifies the channel is clear in its own transmission range, a hidden node may cause a collision at a receiving node. In addition, if two neighbors assess a clear channel and transmit simultaneously, a collision will occur.

Ramifications of this environment are subtle but significant. Unless specific means are implemented at the network layer, a node has no way of knowing whether a packet was successfully reached by its neighbors. In congested networks, a significant amount of collisions occur leading to many dropped packets. The most effective broadcasting protocols try to limit the probability of collisions by limiting the number of rebroadcasts in the network.

2.1.2 Jitter and Random Assessment Delay

As mentioned in Section 2.1.1, 802.11 has no direct means to prevent multiple nodes from assessing a clear channel and broadcasting at the same time. Suppose a source node originates a broadcast packet. Given that radio waves propagate at the speed of light, all neighbors will receive the transmission almost simultaneously. Assuming similar hardware and system loads, the neighbors will process the packet and rebroadcast at the same time. To overcome this problem, broadcast protocols jitter the scheduling of broadcast packets from the network layer to the MAC layer by some uniform random amount of time. This offset allows one neighbor to obtain the channel first, while other neighbors detect that the channel is busy (clear channel assessment fails). The jitter time can be relatively small, just enough to allow one node access to the channel before the others.

Many of the broadcasting protocols require a node to keep track of redundant packets received over a short time interval in order to determine whether to rebroadcast. That time interval, which we have arbitrarily termed "Random Assessment

Delay" (RAD), is randomly chosen from a uniform distribution between 0 and T_{max} seconds, where T_{max} is the highest possible delay interval. This delay in transmission accomplishes two things. First it allows nodes sufficient time to receive redundant packets and assess whether to rebroadcast. Second, the randomized scheduling prevents the collisions described in the jitter discussion.

An important design consideration is the implementation of the random assessment delay. One approach is to send broadcast packets to the MAC layer after a short random time similar to the jitter. In this case, packets remain in the interface queue (IFQ) until the channel becomes clear for broadcast. While the packet is in the IFQ, redundant packets may be received, allowing the network layer to determine if rebroadcasting is still required. If the network layer protocol decides the packet should not be rebroadcast, it informs the MAC layer to discard the packet. Obviously, this approach is only valid if the network layer is able to remove a packet pending in the MAC's IFQ.

A second approach is to implement the random assessment delay as a longer time period and keep the packet at the network layer until the RAD expires. Retransmission assessment is done considering all redundant packets during the RAD. After RAD expiration, the packet is either sent to the MAC layer or dropped. No attempts are made by the network layer to remove the packet after sending it to the MAC layer.

Figure 2.1 is a histogram for average node IFQ lengths given 60 nodes Flooding 80 new packets per second, which is the highest congestion level studied in this thesis (see Section 4.2 for other simulation parameters). From the histogram we see that 42 of 60 nodes (70%) have average queue lengths less than 1 packet. An analysis of the network showed a direct correlation between number of neighbors and queue

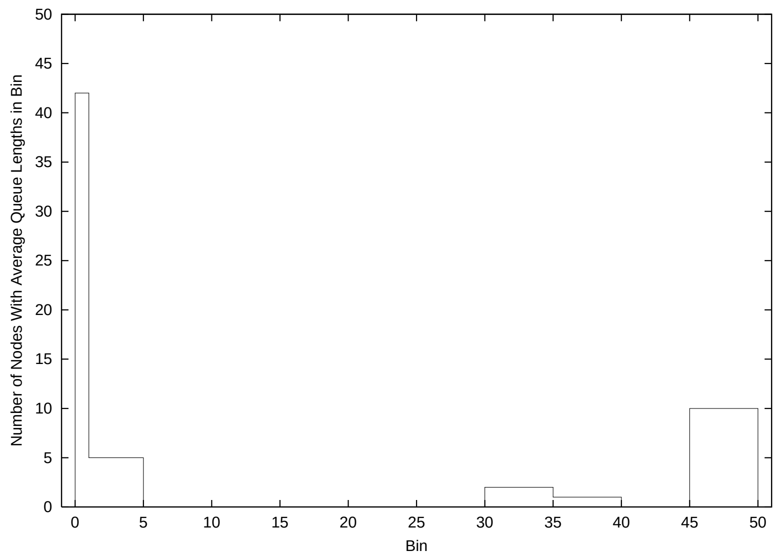

FIG. 2.1. Histogram of Average Interface Queue Lengths for Simple Flooding

length; the nodes with the most neighbors had the longest queues. We surmise that perimeter nodes fail clear channel assessment less often than internal nodes, allowing the perimeter nodes more frequent access to the channel and barraging the interior nodes with traffic. For those perimeter nodes, if the network layer informs the MAC layer to remove a particular packet from its IFQ, it is unlikely the packet will still be pending in the queue. Thus, we chose to implement the RAD via the second approach discussed. We note however, that given the RTS/CTS/ACK procedure required by 802.11 for unicast packets, it is likely that a mixture of unicast and broadcast traffic would produce a more even distribution of queue lengths; that is, unicast packets have more tendency to block at the head of the queue.

2.1.3 Prevention of Infinite Loops

None of the protocols categorized in this thesis require that a node rebroadcast a given packet more than one time. Thus, each broadcast protocol requires that nodes cache the original source node ID of the packet and the packet ID. This allows the protocol to uniquely identify each broadcast packet and assign appropriate behavior upon reception of a packet.

2.2 Categorization of Protocols

Twelve broadcast protocols are described in this section. A comprehensive comparison of these protocols would include implementing each and simulating all twelve under a wide range of parameters. This exercise would require significant work that we did not believe necessary in order to glean an overall understanding of the strengths and weaknesses of the protocols. Instead, our approach was to categorize the protocols into four families: Simple Flooding, Probability Based Methods, Area Based Methods and Neighbor Knowledge Methods. Implementing and evaluating one or two representative protocols from each category allows us to make conclusions regarding the applicability of all protocols within each family. We give an overview of each category in the following paragraph. Details on the protocols within each category are supplied in Sections 2.2.1-2.2.4. Section 2.3 provides a review of relevant protocol features.

Simple Flooding requires each node to rebroadcast all packets. Probability Based Methods use some basic understanding of the network topology to assign a probability to a node to rebroadcast. Area Based Methods assume nodes have common transmission distances; a node will rebroadcast only if the rebroadcast will reach sufficient additional coverage area. Neighbor Knowledge Methods maintain state on their neighborhood, via “Hello” packets, which is used in the decision to rebroadcast. Note that our four categories are presented in order of increasing algorithm complexity and per node state requirement. The goal of the added cost is to reduce the number of redundant transmissions.

2.2.1 Simple Flooding

The algorithm for Simple Flooding [32, 33] starts with a source node broadcasting a packet to all neighbors. Each of those neighbors in turn rebroadcast the packet exactly one time and this continues until all reachable network nodes have received the packet. In [32], Ho et al. propose Flooding as a scheme to achieve reliable broadcast and multicast in highly dynamic networks. [33] is an IETF Internet Draft proposing the use of Flooding as a “Simple Protocol” for broadcasting and multicasting in ad hoc networks which are characterized by low node densities and/or high mobility.

2.2.2 Probability Based Methods

Two protocols are considered in this section, the Probabilistic Scheme and the Counter-Based Scheme. The Probabilistic scheme from [34] is similar to Flooding, except that nodes only rebroadcast with a predetermined probability. In dense networks multiple nodes share similar transmission coverages. Thus, randomly having some nodes not rebroadcast saves node and network resources without harming delivery effectiveness. In sparse networks, there is much less shared coverage; thus, nodes will not receive all the broadcast packets with the Probabilistic scheme unless the probability parameter is high. When the probability is 100%, this scheme is identical to Flooding.

Ni et al [34] show an inverse relationship between the number of times a packet is received at a node and the probability of that node being able to reach additional area on a rebroadcast. This result is the basis of their Counter-Based scheme. Upon reception of a previously unseen packet, the node initiates a counter with a value of one and sets a RAD (which is randomly chosen between 0 and T_{max} seconds). During the RAD, the counter is incremented by one for each redundant packet received. If the counter is less than a threshold value when the RAD expires, the packet is rebroadcast. Otherwise, it is simply dropped. From [34], threshold values above six relate to little additional coverage area being reached.

The overriding compelling features of the Counter-Based scheme are its simplicity and its inherent adaptability to local topologies. That is, in a dense area of the network, some nodes won't rebroadcast; in sparse areas of the network, all nodes rebroadcast.

2.2.3 Area Based Methods

The Distance-Based scheme and the Location-Based scheme are two Area Based methods. Suppose a node receives a packet from a sender that is located only one meter away. If the receiving node rebroadcasts, the additional area covered by the retransmission is quite low. On the other extreme, if a node is located at the boundary of the sender node's transmission distance, then a rebroadcast would reach significant additional area, 61% to be precise [34]. Now envision a neighborhood of nodes. A node using an Area Based Method can evaluate additional coverage area based on all received redundant transmissions. Note that area based methods only consider the coverage area of a transmission; they do not consider whether nodes exist within that area.

Nodes using the Distance-Based scheme compare the distance between themselves and each neighbor node that has previously rebroadcast a given packet. Upon reception of a previously unseen packet, a RAD is initiated and redundant packets are cached. When the RAD expires, all source node locations are examined to see if any node is closer than a threshold distance value. If true, the node doesn't rebroadcast.

The authors of [34] note that one benefit of the Distance-Based protocol is that it is implementable without a Global Positioning System (GPS). Specifically, signal strength can be used to calculate the distance from a transmitting node.

The Location-Based scheme [34] uses a more precise estimation of expected additional coverage area in the decision to rebroadcast. In this method, each node must have the means to determine its own location, e.g. a Global Positioning System (GPS).

Whenever a node originates or rebroadcasts a packet it adds its own location to the header of the packet. When a node initially receives a packet, it notes the location of the transmitting node and calculates the additional coverage area obtainable were it to rebroadcast. If the additional area is less than a threshold value, the node will not rebroadcast, and all future receptions of the same packet will be ignored. Otherwise, the node assigns a RAD before delivery. If the node receives a redundant packet during the RAD, it recalculates the additional coverage area and compares that value to the threshold. The area calculation and threshold comparison occur with all redundant broadcasts received until the packet reaches either its scheduled send time or is dropped.

2.2.4 Neighbor Knowledge Methods

The Neighbor Knowledge methods make up the largest category. The following protocols are discussed:

1. Flooding with Self Pruning
2. Scalable Broadcast Algorithm (SBA)
3. Dominant Pruning
4. Multipoint Relaying
5. Ad Hoc Broadcast Protocol
6. CDS-Based Broadcast Algorithm
7. Lightweight and Efficient Network Wide Broadcast (LENWB) Protocol

The simplest of the Neighbor Knowledge Methods is what Lim and Kim refer to as Flooding with Self Pruning [35]. This protocol requires that each node have knowledge of its 1-hop neighbors. "Hello" packets are periodically broadcast a single hop to let neighbors know that the node exists.

A node includes its list of known neighbors in the header of each broadcast packet. A node receiving a broadcast packet compares its neighbor list to the sender's neighbor list. If the receiving node would not reach any additional nodes, it refrains from rebroadcasting; otherwise the node rebroadcasts the packet.

The Scalable Broadcast Algorithm (SBA) [36] requires that all nodes have knowledge of their neighbors within a two hop radius. This neighbor knowledge coupled with the identity of the node from which a packet is received allows a receiving node to determine if it would reach additional nodes by rebroadcasting. 2-hop neighbor knowledge is achievable via "Hello" packets; each node periodically performs a one hop broadcast sending its own node identifier (IP address) and the list of known neighbors. After a node receives a "Hello" packet from all its neighbors, it has two hop topology information centered at itself.

Suppose Node B receives a broadcast data packet from Node A. Since Node A is a neighbor, Node B knows all of its neighbors, common to Node A, that have also received Node A's transmission of the broadcast packet. If Node B has additional neighbors not reached by Node A's broadcast, Node B schedules the packet for delivery with a RAD. If Node B, while waiting for the RAD to expire, receives a redundant broadcast packet from another neighbor, Node B again determines if it can reach any new nodes by rebroadcasting. This process continues until either the RAD expires and the packet is sent, or the packet is dropped.

The authors of [36] propose a method to dynamically adjust the RAD to network conditions; they weight the time delay based on a node's relative neighbor degree. Specifically, each node searches its neighbor tables for the maximum neighbor degree of any neighbor node, d_{Nmax}. It then calculates a RAD based on the ratio given by:

$$\left(\frac{d_{Nmax}}{d_{me}}\right), \tag{2.1}$$

where d_{me} is the node's current number of neighbors. The weighting scheme shown in Equation 2.1 is greedy; nodes with the most neighbors usually broadcast before the others.

Dominant Pruning [35] also uses 2-hop neighbor knowledge, obtained via "Hello" packets, for routing decisions. Unlike SBA, however, Dominant Pruning requires rebroadcasting nodes to proactively choose some or all of its 1-hop neighbors as rebroadcasting nodes. Only those chosen nodes are allowed to rebroadcast. Nodes inform neighbors that they must rebroadcast by including their address as part of a list in each broadcast packet header.

When a node receives a broadcast packet it checks the header to see if its address is part of the list. If so, it uses a Greedy Set Cover algorithm to determine which subset of neighbors should rebroadcast the packet, given knowledge of which neighbors have already been covered by the sender's broadcast. The Greedy Set Cover algorithm, as adapted in [35] from [37], recursively chooses 1-hop neighbors which cover the most 2-hop neighbors and recalculates the cover set until all 2-hop neighbors are covered.

Multipoint Relaying [38] is similar to Dominant Pruning in that rebroadcasting nodes are explicitly chosen by upstream senders. For example, say Node A is originating a broadcast packet. It has previously selected some, or in certain cases all, of it one hop neighbors to rebroadcast all packets they receive from Node A. The chosen nodes are called Multipoint Relays (MPRs) and they are the only nodes allowed to rebroadcast a packet received from Node A. Each MPR is required to choose a subset of its one hop neighbors to act as MPRs as well.

Since a node knows the network topology within a 2-hop radius, it can select 1-hop neighbors as MPRs that most efficiently reach all nodes within the two hop neighborhood. The authors of [38] propose the following algorithm for a node to choose its MPRs:

1. Find all 2-hop neighbors that can only be reached by one 1-hop neighbor. Assign those 1-hop neighbors as MPRs.

2. Determine the resultant cover set (i.e., the set of 2-hop neighbors that will receive the packet from the current MPR set).

3. From the remaining 1-hop neighbors not yet in the MPR set, find the one that would cover the most 2-hop neighbors not in the cover set. Assign this 1-hop neighbor as an MPR.

4. Repeat from step 2 until all 2-hop neighbors are covered.

Multipoint Relaying is described in detail as part of the Optimized Link State Routing (OLSR) protocol defined by an Internet draft [39]. In this implementation, “Hello” Packets include fields for a node to list the MPRs it has chosen. Anytime a node receives a “Hello” packet, it checks if it is a MPR for the source of the packet. If so, it must rebroadcast all data packets received from that source. Clearly, the update interval for “Hello” packets must be carefully chosen and, if possible, optimized for network conditions.

The Ad Hoc Broadcast Protocol (AHBP) [40] utilizes an approach similar to Multipoint Relaying. In AHBP, only nodes who are designated as a Broadcast Relay Gateway (BRG) within a broadcast packet header are allowed to rebroadcast the packet. BRGs are proactively chosen from each upstream sender which is a BRG itself. The algorithm for a BRG to choose its BRG set is identical to that used in Multipoint Relaying (see steps 1-4 for choosing MPRs).

AHBP differs from Multipoint Relaying in three ways:

1. A node using AHBP informs 1-hop neighbors of the BRG designation within the header of each broadcast packet. This allows a node to calculate the most effective BRG set at the time a broadcast packet is transmitted. In contrast, Multipoint Relaying informs 1-hop neighbors of the MPR designation via “Hello” packets.

2. In AHBP, when a node receives a broadcast packet and is listed as a BRG, the node uses 2-hop neighbor knowledge to determine which neighbors also received the broadcast packet in the same transmission. These neighbors are considered already “covered” and are removed from the neighbor graph used to choose

next hop BRGs. In contrast, MPRs are not chosen considering the source route of the broadcast packet. MPR assignment is performed at the "Hello" packet interval and therefore must be independent of any broadcast source.

3. AHBP is extended to account for high mobility networks. Suppose Node A receives a broadcast packet from Node B, and Node A does not list Node B as a neighbor (i.e., Node A and Node B have not yet exchanged "Hello" packets). In AHBP-EX (extended AHBP), Node A will assume BRG status and rebroadcast the node. Multipoint relaying could be similarly extended.

Peng and Lu describe the Connected Dominating Set (CDS)-Based Broadcast Algorithm, a more calculation intensive algorithm for selecting BRGs, in [41]. While AHBP only considers the source of the broadcast packet to determine a receiving node's initial cover set, CDS-Based Broadcast Algorithm *also* considers the set of higher priority BRGs selected by the previous sender [41]. For example, suppose Node A has selected Nodes B, C and D (in this order) to be BRGs. When Node C receives a broadcast packet from Node A, AHBP requires Node C to add neighbors common to Node A to the initial cover set. CDS-Based Broadcasting *also* requires that Node C adds neighbors common to Node B, because Node B is a higher priority BRG. Likewise, Node D is required to consider common neighbors with nodes A, B and C.

Once the initial cover set is determined, a node then chooses which neighbors should function as BRGs. The algorithm for determining this is the same as that for AHBP and Multipoint Relaying (see steps 1-4 for choosing Multipoint Relays).

The Lightweight and Efficient Network-Wide Broadcast (LENWB) protocol [42] also relies on 2-hop neighbor knowledge obtained from "Hello" packets. However, instead of a node explicitly choosing nodes to rebroadcast, the decision is implicit. In

LENWB, each node decides to rebroadcast based on knowledge of which of its other one and two hop neighbors are expected to rebroadcast. The information required for that decision is knowledge of which neighbors have received a packet from the common source node and which neighbors have a higher priority for rebroadcasting. The priority is proportional to a node's number of neighbors; the higher the node's degree the higher the priority. Since a node relies on its higher priority neighbors to rebroadcast, it can proactively compute if all of its lower priority neighbors will receive those rebroadcasts; if not, the node rebroadcasts.

2.3 Review of Protocol Features

The broadcasting protocols described in this chapter share features that directly affect both applicability and/or node cost. Table 2.1 provides a synopsis of which protocols are characterized by the following important features. Protocols with a RAD require that packets be cached at the node until the RAD expires, which increases memory requirements of the nodes. The Location-Based scheme requires the use of a GPS which is an expensive option, unless the cost of the GPS can be partially absorbed by other applications on the node.

Protocols that require "Hello" packets may be more applicable to networks with a lot of broadcast traffic; otherwise, the "Hello" packet transmissions may be unnecessary overhead. Neighbor knowledge protocols also have more complex algorithms (more than linear time in terms of number of neighbors).

As we'll see from simulation results in Chapter 4, Neighbor Knowledge protocols that adjust cover sets, based on the assumption that 2-hop neighbors have been previously reached by other senders (Dominant Node Based Protocols), perform poorly in certain conditions. Lastly, Multipoint Relaying is alone in its proactive approach to maintaining MPR designation; this proactive behavior may prove inefficient in low broadcast traffic networks. All protocols are more closely analyzed in Chapter 3. In Chapter 4, simulation results are given for a subset (at least one from each category) of the broadcast protocols.

Protocol	**RAD/ Jitter**	**GPS**	**Hello Packets**	**Complex Algorithm**	**Dominant Node Based**	**Reactive/ Proactive**
Simple Flooding	Jitter	No	No	No	No	Reactive
Probabilistic	Jitter	No	No	No	No	Reactive
Counter-Based	RAD	No	No	No	No	Reactive
Distance-Based	RAD	Yes/No	No	No	No	Reactive
Location-Based	RAD	Yes	No	No	No	Reactive
Flooding w/ Self Pruning	Jitter	No	Yes	No	No	Reactive
SBA	RAD	No	Yes	Yes	No	Reactive
Dominant Pruning	Jitter	No	Yes	Yes	Yes	Reactive
Multipoint Relaying	Jitter	No	Yes	Yes	Yes	Proactive
AHBP	Jitter	No	Yes	Yes	Yes	Reactive
CDS-Based	Jitter	No	Yes	Yes	Yes	Reactive
LENWB	Jitter	No	Yes	Yes	Yes	Reactive

Table 2.1. Protocol Features

Chapter 3

THEORY-BASED COMPARISON OF BROADCAST PROTOCOLS

3.1 Background

Researchers often use simulators to evaluate MANET routing protocols. Usually, the entire protocol stack is simulated so that researchers can infer, with some level of confidence, how protocols respond to constraints inherent in "real" networks. In other words, the complexities of protocol layer interactions in dynamically configurable networks make explicit mathematical determination of performance prohibitively difficult, and hence the reliance on simulation. Unfortunately, simulation does not always provide fundamental insight into protocol issues. For example, simulations on lab PCs may not capture CPU cycle constraints inherent in actual mobile devices in specific types of networks (i.e., sensor networks).

Additionally, protocol simulation performance results at specific network conditions may not apply to different conditions. For example, we make conclusions regarding protocol applicability that is inferred from our simulation comparisons and is based on the assumption that higher complexity algorithms pay for themselves. Specifically, we assume that CPU cycles are much cheaper than packet transmissions in terms of hardware cost and power consumption (work in [43] supports our assumptions). While the cost assumption may be valid for many network conditions, it is not valid for all. For example, in a sparse network where each node rebroadcasts each broadcast packet, it does not pay to incur the cost of a complex algorithm; in this case, Simple Flooding is appropriate.

The point is that simulation often provides an incomplete view of protocol performance and interactions. Therefore, in this chapter, we provide two theory-based comparisons of broadcasting protocols in order to have a richer understanding of them. In Section 3.2, we consider the run time complexity of different operations of each protocol (i.e., duplicate packet detection, beaconing, and retransmission determination). Since this analysis considers the routing protocol isolated from the protocol stack, it becomes possible for future implementors to gauge applicability for their specific network environment. In Section 3.3 we develop a mathematical model for predicting the retransmission frequency of nodes using two of the broadcasting protocols. The benefit of the model is that it allows us to pinpoint network parameters that most affect the protocols.

3.2 Algorithm Complexities

Broadcasting protocol algorithms contain three fundamental parts:

1. Duplicate Packet Detection and Table Maintenance,
2. Neighbor Table Maintenance, and
3. Core Algorithm for Retransmission Determination.

Items 1 and 3 apply to all broadcasting protocols. Item 2 applies only to Neighbor Knowledge methods. We discuss each item in a subsection below. In all cases we develop a worst case run time analysis. We note that tighter worst case bounds may exist.

Analysis of algorithm complexity requires some assumptions regarding the underlying data structures. We assume that the implementation of each item above utilizes a balanced search tree (e.g., red-black tree or B-tree). Further, for discussions regarding neighbor table maintenance and retransmission determination, we need a worst case estimate for the number of entries in the neighbor tables and associated structures. If γ_{max} is the maximum number of neighbors in a given network, one might expect γ_{max} elements to be an upper bound for the number of table entries. In practice, this does not account for the situation in which entries are kept in the table for a period longer than the beaconing interval. Nor does it account for single periods in which new entries are added before out-of-date information is pruned. However, instead of attempting to quantify a value that is application specific, we assume γ_{max} is the maximum number of neighbors and the maximum number of entries in a neighbor table.

3.2.1 Duplicate Packet Detection and Table Maintenance

To maintain unique packet identifiers, each packet ID is a tuple of the originating node ID followed by a unique sequence number. This method obviates the need for distributed agreement of globally unique packet IDs. Figure 3.1 shows the first few elements of the data structure that each node uses to maintain information on received packets. Each tree element contains a key (network node ID) and a pointer to a second balanced tree of unique packet sequence numbers received from this network node. In order to perform a worst case run time analysis of determining if a packet is a duplicate, we assume that each network node has a tree element in the primary tree of Figure 3.1. Therefore, the primary tree has N elements, where N is the number of nodes in the network. A search for a node ID in the primary tree is bounded by $\log N$ or $O(\log N)$. Similarly, if we assume that one node originates all broadcast packets (designated as B_o), a search of that element's secondary tree is $O(\log B_o)$. If the sequence number is not found, then it must be added to the balanced tree; the

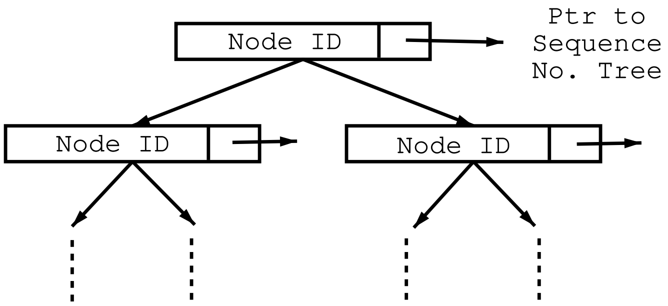

FIG. 3.1. Received Packet Table

insertion is also $O(\log B_o)$. Thus, the complexity for duplicate packet detection and table maintenance is $O(\log N + 2\log B_o)$ which reduces to:

$$O(\log N + \log B_o). \tag{3.1}$$

3.2.2 Neighbor Table Maintenance

Recall that Flooding with Self Pruning uses 1-hop neighbor knowledge in the decision on whether to rebroadcast. Let us suppose that the neighbor table for a node using Flooding with Self Pruning is a balanced search tree, such that each element in the tree contains a field for neighbor ID (key) and timestamp (TS) as shown in Figure 3.2. Two operations must be considered for maintenance of this neighbor table. The neighbor table must be updated when a node receives a "Hello" packet and out-of-date data must be removed from the table on a periodic basis.

First consider the case of a received "Hello" packet. Upon reception, the receiving node first searches for the neighbor ID in the neighbor table. If the ID exists, the timestamp is updated to reflect the new reception time. If the ID does not exist, an entry is added to the tree. For the worst case run time analysis, the node ID does not exist in the tree. Given a maximum of γ_{max} table entries, both an ID

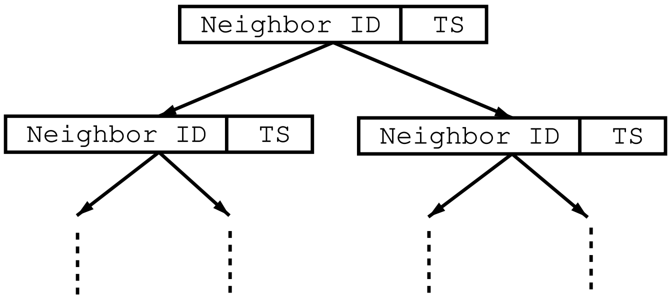

FIG. 3.2. 1-Hop Neighbor Table

search and an ID insertion is bounded by $O(\log \gamma_{max})$. The combined operation of ID search followed by ID insertion is $O(2 \log \gamma_{max})$. Since a node can expect to receive γ_{max} "Hello" packets (one from each neighbor), there will be γ_{max} search/insertion operations during each beaconing interval. In other words, updating the neighbor table costs $O(2\gamma_{max} \log \gamma_{max})$ operations per beaconing interval.

To remove out-of-date data from the tree, each element TS in the tree must be inspected which costs $O(\gamma_{max})$. When an obsolete tree element is identified, it must be deleted; the deletion operation costs $O(\log \gamma_{max})$ time. In the worst case, all γ_{max} tree elements are found to be obsolete each beaconing period and must be removed. Thus, pruning out-of-date information is $O(\gamma_{max} + \gamma_{max} \log \gamma_{max})$.

The entire procedure to maintain the neighbor table has a worst case complexity of $O(\gamma_{max} + 3\gamma_{max} \log \gamma_{max})$, which reduces to

$$O(\gamma_{max} \log \gamma_{max}),$$

for each "Hello" interval.

All Neighbor Knowledge methods (except Flooding with Self Pruning) require a node to maintain a 2-hop neighbor table. Figure 3.3 shows the tree representation of the 2-hop neighbor table that we assume. Each tree element contains a field for the

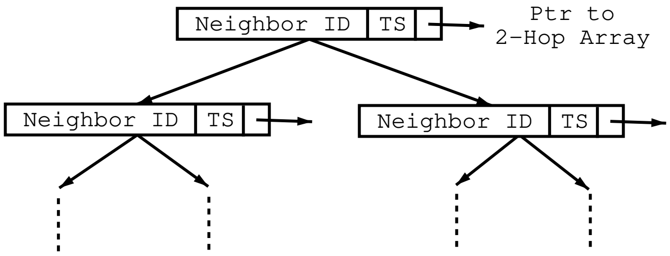

FIG. 3.3. 2-Hop Neighbor Table

1-hop neighbor ID, a timestamp, and a pointer to an array structure which contains an ordered list of 2-hop neighbor IDs. When a "Hello" packet is received, the neighbor table is searched for the sender's ID. If found, the timestamp and the entire 2-hop neighbor array are updated. If the sender's ID is not found, a new element is inserted in the tree. Similar to the maintenance of a 1-hop neighbor table, the worst case occurs when an insertion step is required. The search/insertion operation for all γ_{max} neighbors has a complexity of $O\left(2\gamma_{max}\log\gamma_{max}\right)$. If we assume that each neighbor sends its neighbor list as an ordered list and that each neighbor has γ_{max} neighbors of its own, then the complexity of creating a 2-hop array for one neighbor is γ_{max} and for all γ_{max} neighbors is $O\left(\gamma_{max}^2\right)$.

Pruning out-of-date information in 2-hop neighbor tables has the same complexity as that for 1-hop neighbor tables; it is $O\left(\gamma_{max} + \gamma_{max}\log\gamma_{max}\right)$. Thus, the entire 2-hop neighbor table maintenance complexity is

$$O\left(\gamma_{max}^2 + \gamma_{max} + 3\gamma_{max}\ \log\gamma_{max}\right)$$

which reduces to

$$O\left(\gamma_{max}^2\right).$$

Algorithm Step	**Complexity**
Find all 2-hop neighbors only reachable by one 1-hop neighbor. Designate these 1-hop neighbors as MPRs. To accomplish, build a second tree data structure, indexed by 2-hop neighbor IDs, which point to an array of 1-hop neighbor IDs that can reach this 2-hop neighbor.	$O\left(\gamma_{max}^2 \log \gamma_{max}\right)$
Determine the resultant cover set.	$O\left(\gamma_{max}^2\right)$
Choose a remaining 1-hop neighbor that reaches the most uncovered 2-hop neighbors. Designate it as an MPR.	$O\left(\gamma_{max}\right)$
Determine the resultant cover set.	$O\left(\gamma_{max}^2\right)$
Iterate through previous two steps until all 2-hop nodes are covered.	$O\left(\gamma_{max}\right)$ iterations
Complexity of MPR selection	$O\left(\gamma_{max}^3\right)$

Table 3.1. Algorithm Complexity of Multipoint Relay Selection in Multipoint Relaying

The Multipoint Relaying protocol provides additional periodic functionality to Neighbor Table updates. In addition to maintaining up-to-date 2-hop neighbor information, Multipoint Relaying also designates MPRs among its 1-hop neighbors. The MPR algorithm determines which neighbors will rebroadcast a node's packets (see Section 2.2.4). Table 3.1 lists algorithm steps, complexity for each step and an overall algorithm complexity for determining a node's MPRs. The rest of this section is devoted to discussing the details of Table 3.1.

Figure 3.4 shows the tree structure that we assume is created in the first step of Table 3.1. The tree in Figure 3.4 is indexed by 2-hop node IDs and each tree element contains a second field which points to another tree of node IDs for each 1-hop node capable of reaching this 2-hop node. To create this structure, we must iterate through each 2-hop neighbor in the 2-hop neighbor arrays (see Figure 3.3). If

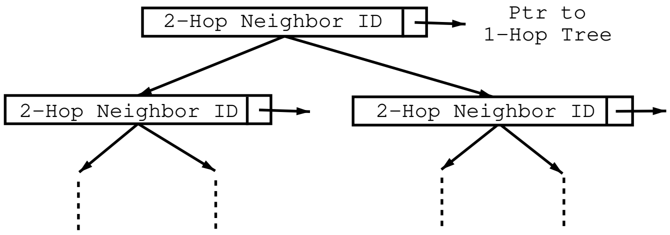

FIG. 3.4. Second Linked List Structure for Multipoint Relay Selection

we assume γ_{max} 2-hop neighbors for each of the γ_{max} 1-hop neighbors, then $O\left(\gamma_{max}^2\right)$ iterations are needed to iterate through all 2-hop neighbor arrays. In addition, each iteration requires a $O\left(\log \gamma_{max}^2\right)$ search to determine if the 2-hop neighbor ID is already in the new structure. If it is, only an $O\left(\log \gamma_{max}\right)$ insertion operation is required to add the 1-hop "parent" of this node to the 1-hop tree. If the 2-hop neighbor ID is not in the new structure, a $O\left(\log \gamma_{max}^2\right)$ insertion operation is first required and then an $O(1)$ insertion operation is required to add the 1-hop "parent" of this node to the 1-hop tree. Therefore, constructing Figure 3.4 has a worst case complexity of $O\left(\gamma_{max}^2\left(\log \gamma_{max}^2 + \log \gamma_{max}^2 + 1\right)\right)$. If we implement each 2-hop tree element in Figure 3.4 with a field for the number of 1-hop nodes that can reach this node, then a simple tree traversal of 2-hop neighbor ID nodes will suffice to find those 2-hop neighbors only reachable by one 1-hop neighbor. Thus, the complexity of Step 1 in Table 3.1 is the highest order term from

$$O\left(\gamma_{max}^2\left(\log\ \gamma_{max}^2 + \log \gamma_{max}^2 + 1\right) + \gamma_{max}^2\right),$$

which reduces to $O\left(\gamma_{max}^2 \log \gamma_{max}\right)$.

In the second step of Table 3.1 we find the resultant cover set. Since the 2-hop neighbors reachable by only one 1-hop neighbor have been identified, and since those 1-hop neighbors have been designated as MPRs, we must remove from consideration all 2-hop neighbors that can be reached by these 1-hop neighbors from the trees in Figure 3.3 and Figure 3.4. (Instead of removing elements from the tree, which is costly, we set a bit in the tree element to indicate it has been "covered".) If we assume both trees are constructed with mutually referencing pointers (i.e., 1-hop neighbor elements in Figure 3.4 point to 1-hop neighbor elements in Figure 3.3 and 2-hop neighbor elements in Figure 3.3 point to 2-hop neighbor elements in Figure 3.4), we can iterate through each MPR's 2-hop neighbors in Figure 3.3 and then use Figure 3.4 to navigate to all 1-hop neighbors in Figure 3.3 that reach these 2-hop neighbors. In other words, these 2-hop neighbors can simply be designated as "covered" in both structures. Since at most γ_{max} 1-hop neighbors can be designated as MPRs, and since each of these 1-hop neighbors can have at most γ_{max} 2-hop neighbors, the worst case cost to navigate through all of these 2-hop neighbors is $O\left(\gamma_{max}^2\right)$. The complexity of the fourth step in Table 3.1 is derived similarly.

In step three of Table 3.1, we designate the 1-hop neighbor that reaches the most 2-hop neighbors as an MPR. This requires a simple linear time traversal of 1-hop neighbors in Figure 3.3, since we assume each 1-hop tree element includes a field for the number of "uncovered" 2-hop nodes that this node can reach. Thus, this step has a complexity of $O(\gamma_{max})$.

Theoretically, no more than γ_{max} iterations of steps 3 and 4 can exist. These γ_{max} iterations, therefore have a total cost of $O(\gamma_{max}\left(\gamma_{max}^2 + \gamma_{max}\right))$. Since $O\left(\gamma_{max}^3\right)$ is the highest order term in the complexity of the algorithm, we adopt it as the complexity of MPR selection.

3.2.3 Core Algorithm for Rebroadcast Determination

Simple Flooding, Probabilistic, Counter-Based and Distance-Based broadcasting schemes require constant time algorithms for the determination to rebroadcast. Simple Flooding simply rebroadcasts every packet. The Probabilistic scheme generates a random number and compares its value against a threshold to determine whether to rebroadcast. The Counter-Based and Distance-Based schemes compare a value (a counter or a minimum distance) against a threshold to determine whether to rebroadcast.

The Location-Based scheme uses a convex polygon approximation method to determine if the node will reach sufficient additional area in a rebroadcast. The method is based on summing the angles formed by the node (or vertex) and each pair of neighbor nodes from which the packet has been received (points on the polygon). If the sum is 2π then the node lies inside the polygon and is unlikely to reach sufficient additional area to warrant a rebroadcast. If the sum is 0, then the node lies outside the polygon. Since the number of angles is equal to the number of neighbors, the polygon approximation has a complexity of $O(\gamma_{max})$, if we assume the maximum number of neighbors has rebroadcast before the node's RAD has expired. Thus, the complexity of determining whether to rebroadcast for the Location-Based scheme is

$$O(\gamma_{max})$$

for each unique received packet.

Upon reception of a broadcast packet, a node utilizing the Flooding with Self Pruning protocol will iterate through each node ID in the received packet header to assess whether each of its neighbors has received the packet. Therefore, there are a maximum of γ_{max} searches, each requiring $\log \gamma_{max}$ operations. In other words, the complexity of determining whether to rebroadcast for Flooding with Self Pruning is

$$O\left(\gamma_{max} \ \log \gamma_{max}\right)$$

for each unique received packet.

An SBA node that receives a packet searches to see if any of the source's 1-hop neighbors are also 1-hop neighbors of itself (each search runs in $\log \gamma_{max}$ time). If we assume the sender has the maximum number of neighbors, then we can expect γ_{max} searches. Thus, the total complexity per received packet is $O\left(\gamma_{max} \ \log \gamma_{max}\right)$. It is possible for an SBA node to receive duplicate packets from each of its neighbors before its RAD expires. If we assume the worst case of γ_{max} neighbors, then the complexity of determining whether to rebroadcast in SBA is

$$O\left(\gamma_{max}^2 \ \log \gamma_{max}\right)$$

for each received unique packet.

Packets broadcast with Dominant Pruning, AHBP, CDS and LENWB all use a set covering algorithm similar in complexity to the third and fourth steps of the MPR selection in Multipoint Relaying (see Section 3.2.2). Since these protocols may also require γ_{max} iterations of the set covering algorithm, the reduced complexity of each of these protocols is

$$O\left(\gamma_{max}^3\right).$$

Note that the complexity of the neighbor knowledge methods is identical to the complexity of neighbor table maintenance in Multipoint Relaying. For Multipoint Relaying, rebroadcasting nodes are selected during each beaconing period. When a node receives a broadcast packet, it searches its 1-hop neighbor table list to determine whether it is an MPR for the source. Thus, the complexity of determining whether to rebroadcast for the Multipoint Relaying protocol is

$$O\left(\log \gamma_{max}\right) \tag{3.2}$$

for each received unique packet.

3.2.4 Review of Complexity

A side by side comparison of the broadcasting protocol complexities is provided in Table 3.2. Duplicate packet detection must be performed for every broadcast packet received and is therefore a function of both the network wide origination rate of broadcast packets *and* the number of neighbors that rebroadcast a redundant packet. Nodes using Simple Flooding can expect to receive redundant broadcast packets from each neighbor. Efficient broadcast protocols minimize redundant retransmissions; thus, the actual run time cost for duplicate packet detection in efficient broadcast protocols will be less than that incurred by Simple Flooding.

The cost of Neighbor Table maintenance is incurred one time for each beaconing interval. The Neighbor Knowledge methods must perform this operation regardless of the amount of broadcast traffic. (We recommend an adaptive method in Section 4.3.2, which minimizes this cost based on the amount of broadcast traffic *and* the amount of change in network topology.) As shown in Table 3.2, the complexity of neighbor table maintenance for Flooding with Self Pruning is the least, since it only uses 1-hop neighbor knowledge. The complexity of neighbor table maintenance for Multipoint

Relaying is the most, since it designates MPRs proactively.

Rebroadcast determination is performed for each non-duplicate packet received; the frequency of this determination is a function of the broadcast packet origination rate. With the exception of Multipoint Relaying, which proactively determines MPR designation at each beaconing period, the Neighbor Knowledge protocols incur high algorithmic costs for rebroadcast determination. In situations of a sufficiently high broadcast packet origination rate and a correspondingly low beaconing interval, Multipoint Relaying is (overall) less costly than the other Neighbor Knowledge methods. These other Neighbor Knowledge methods, however, are preferred over Multipoint Relaying when the broadcast packet origination rate is low.

Protocol	**Duplicate Packet Detection**	**Neighbor Table Maintenance**	**Rebroadcast Determination**
Simple Flooding	$O(\log N + \log B_o)$	-	$O(1)$
Probability-Based	$O(\log N + \log B_o)$	-	$O(1)$
Counter-Based	$O(\log N + \log B_o)$	-	$O(1)$
Distance-Based	$O(\log N + \log B_o)$	-	$O(1)$
Location-Based	$O(\log N + \log B_o)$	-	$O(\gamma_{max})$
Flood with Self Pruning	$O(\log N + \log B_o)$	$O(\gamma_{max} \log \gamma_{max})$	$O(\gamma_{max} \log \gamma_{max})$
SBA	$O(\log N + \log B_o)$	$O(\gamma_{max}^2)$	$O(\gamma_{max}^2 \log \gamma_{max})$
Dominant Pruning	$O(\log N + \log B_o)$	$O(\gamma_{max}^2)$	$O(\gamma_{max}^3)$
Multipoint Relaying	$O(\log N + \log B_o)$	$O(\gamma_{max}^3)$	$O(\log \gamma_{max})$
AHBP	$O(\log N + \log B_o)$	$O(\gamma_{max}^2)$	$O(\gamma_{max}^3)$
CDS-Based	$O(\log N + \log B_o)$	$O(\gamma_{max}^2)$	$O(\gamma_{max}^3)$
LENWB	$O(\log N + \log B_o)$	$O(\gamma_{max}^2)$	$O(\gamma_{max}^3)$

Table 3.2. Run Time Complexities of Broadcasting Protocols

3.3 Prediction of Retransmission Frequency

In this section we propose a model for predicting retransmission frequency for two of the broadcasting protocols, the Counter-Based scheme and the Distance-Based scheme. We consider these protocols under idealized conditions of a Null MAC and static topology to ensure that the other protocol layers do not interfere with the study. We also only consider fully connected networks. The benefit of developing this model is that it allows us to discover which protocol parameters and network conditions affect retransmission frequency. The prediction model may also be a viable alternative to time consuming simulation development, especially for networks characterized by low bandwidth congestion and infrequent topology changes.

Our prediction model is probability based and it applies only to networks in which nodes are randomly dispersed and sources are randomly chosen. Our model is not applicable to bandwidth limited networks in which the MAC scheme affects packet delivery.

Our approach is to sum the individual probability of each condition that would induce a node to retransmit a packet. The result of that summation is equivalent to predicting how frequently a node retransmits a broadcast packet given such random conditions.

We can model each broadcasting protocol with Equation 3.3, where R_{node} is the expected number of retransmissions (or rebroadcasts) per node per time period, given S. S is the number of broadcast packets originated during the time period and P is the probability that a given node will rebroadcast a given packet:

$$R_{node} = P * S. \tag{3.3}$$

In Simple Flooding, each node retransmits each broadcast packet once. In other words, $P = 1.0$. Thus, the number of retransmissions per node per time period is

$$R_{node} = S.$$

In sparsely populated networks, nodes using efficient broadcasting schemes may also retransmit all originated messages (i.e., P = 1.0). As neighbor density increases, P is reduced in the efficient broadcasting schemes. That is, P is inversely related to neighbor density. The following two subsections describe our work in mathematically predicting P as a function of neighbor density for the Counter-Based and the Distance-Based schemes. Our approach is to first calculate the individual probabilities for cases which dictate that a node rebroadcast. The summation of these individual probabilities is then P.

3.3.1 Predicting P for the Counter-Based Scheme

In the Counter-Based scheme, a node v will only rebroadcast if it receives fewer than a threshold (T) number of redundant packets before its RAD expires. There are three scenarios in which node v would receive fewer than T redundant packets:

1. Node v has fewer than T neighbors; therefore, it is impossible for v to receive T (or more) copies of the broadcast packet.

2. Node v has T or more neighbors, but fewer than T neighbors rebroadcast the packet. (Recall that v's neighbors are also operating under the same protocol as v.)

3. Node v has T or more neighbors that rebroadcast, but fewer than T neighbors rebroadcast *before* v's RAD expires. (The decision to rebroadcast is made based

on the broadcasts v has received when its RAD timer expires.)

Given these scenarios, let their probabilities of occurrence be P_1, P_2 and P_3 respectively. Thus, for the Counter-Based scheme, $P = P_1 + P_2 + P_3$.

To determine P_1, we first consider a node w which is in the network, is not node v (the assessing node) and is not node u (the node from which v first received the packet). Since nodes are distributed in a uniform and random manner, the probability that w is a neighbor of v, P_n, can be defined as

$$P_n = \frac{\pi r^2}{A}, \tag{3.4}$$

where r is the transmission range of the nodes and A is the area of the entire network. Since each node in the network (except v and u) has the same probability of being a neighbor of v, and since node placement is uniform and random, the number of nodes (excluding v and u) that are neighbors of v (I) follows a binomial distribution:

$$I \sim Bin\left(N-2, P_n\right),$$

where N is the number of nodes in the network. The probability that I is equal to a given value is:

$$P[I = i] \;=\; \binom{N-2}{i} \left(P_n\right)^i \left(1 - P_n\right)^{N-2-i}. \tag{3.5}$$

Since v first received the packet from u, P_1 is really the probability that v has less than $(T-1)$ additional neighbors:

$$P_1 = P[I < T - 1],$$

From Equation 3.5, we then derive:

$$P_1 = \sum_{i=0}^{T-2} \binom{N-2}{i} (P_n)^i (1-P_n)^{N-2-i}. \tag{3.6}$$

To compute P_2, we must calculate the probability that node v has $(T-1)$ or more neighbors but fewer than $(T-1)$ neighbors rebroadcast. We assume that each node broadcasts with a probability P. (We note that this is precisely the quantity that we seek to compute.) The probability that M neighbors of node v rebroadcast is dependent on v's number of neighbors, I. Assuming M follows a binomial distribution, we can represent this conditional probability as:

$$P[M = m \mid I = i] \;=\; \binom{i}{m} (P)^m (1-P)^{i-m}. \tag{3.7}$$

P_2, is the conditional probability that m is less than $(T-1)$ given that I is greater than or equal to $(T-1)$. In other words,

$$P_2 = P[M < T-1 \mid I \geq T-1]$$

or

$$P_2 = \sum_{i=T-1}^{N-2} \left(\binom{N-2}{i} (P_n)^i (1-P_n)^{N-2-i} * \sum_{m=0}^{T-2} \binom{i}{m} (P)^m (1-P)^{i-m} \right). \tag{3.8}$$

In the final scenario, node v rebroadcasts because fewer than T neighbors have broadcast before v's RAD expired. For this scenario to occur independently from the other scenarios, three parameters must be concurrently met:

1. $B < (T-1)$, where B is the number of neighbors that rebroadcast before v.

Since we must account for the initial packet reception from u, B can not be equal to $(T-1)$.

2. $M \geq (T-1)$, where M is the number of neighbors that rebroadcast the packet.

3. $I \geq (T-1)$, where I is node v's number of neighbors.

Consider the case in which v has M neighbors that rebroadcast. We must determine the probability that fewer than $(T-1)$ broadcast before node v does. Since each node assigns a random assessment delay from a uniform and random distribution, and since we assume that all of v's neighbors receive the packet at the same time, then each rebroadcasting neighbor, M, has a 50% probability of rebroadcasting before any other neighbor. Thus, the probability that $B = b$ depends on the value of M:

$$P[B = b \mid M = m] \;=\; \binom{m}{b} (0.5)^b (1-0.5)^{m-b} .$$

If B is any value between 0 and $(T-2)$, inclusive, then node v will rebroadcast. To account for each such case, we sum each probability:

$$P[B < T-1 \mid M = m] \;=\; \sum_{b=0}^{T-2} \binom{m}{b} (0.5)^m .$$

As discussed for P_2, the probability that $M = m$ depends on the value of I. In other words, Equation 3.7 is valid here as well. However, M must be greater than or equal to $(T-1)$. In fact, M can have any value between $(T-1)$ and I. To account for each possibility, we sum each probability:

$$P[M \geq T-1 \mid I = i] \;=\; \sum_{m=T-1}^{i} \binom{i}{m} (P)^m (1-P)^{i-m} .$$

Finally, there is a probability associated with v's number of neighbors. As listed for previous scenarios, there is a possibility that I may be one value from a range of values. We account for each possibility by summing the probabilities of each:

$$P[I \geq T-1] \;=\; \sum_{i=T-1}^{N-2} \binom{N-2}{i} (P_n)^i \, (1-P_n)^{N-2-i}.$$

P_3 is then the probability that $B < (T-1)$ and $M \geq (T-1)$ and $I \geq (T-1)$:

$$\begin{aligned} P_3 = \sum_{i=T-1}^{N-2} \Bigg[\binom{N-2}{i} (P_n)^i \, (1-P_n)^{N-2-i} \\ * \sum_{m=T-1}^{i} \left(\binom{i}{m} (P)^m \, (1-P)^{i-m} \right. \\ \left. * \sum_{b=0}^{T-2} \binom{m}{b} (0.5)^m \right) \Bigg]. \end{aligned} \tag{3.9}$$

Recall that $P = P_1 + P_2 + P_3$. Therefore, using Equations 3.6, 3.8, and 3.9, we have:

$$\begin{aligned} P = \sum_{i=0}^{T-2} \binom{N-2}{i} (P_n)^i \, (1-P_n)^{N-2-i} \\ + \sum_{i=T-1}^{N-2} \Bigg[\binom{N-2}{i} (P_n)^i \, (1-P_n)^{N-2-i} \\ * \left(\sum_{m=0}^{T-2} \binom{i}{m} (P)^m \, (1-P)^{i-m} \right. \\ + \sum_{m=T-1}^{i} \left(\binom{i}{m} (P)^m \, (1-P)^{i-m} \right. \\ \left.\left. * \sum_{b=0}^{T-2} \binom{m}{b} (0.5)^m \right) \right) \Bigg]. \end{aligned} \tag{3.10}$$

We used Matlab software to iteratively find the value of P satisfying Equation 3.10. In order to test our model, we compared it to a simulation of the Counter-

Based scheme in a 350x350 meter network containing a varying number of nodes. We assured that each simulation topology was completely connected by constructing a multihop connectivity matrix from a simple adjacency matrix, and verifying a path existed between all node pairs in the network. We simulated each node with a transmission radius of 100 meters and we simulated the protocol with a Null MAC; thus, each node received each broadcast packet at least one time. Each simulation result presented in Figure 3.5 is the mean and 95% confidence interval for ten different topologies. Each simulation run included 100 broadcast packets transmitted from randomly chosen sources. Since a node at the perimeter of the network may have fewer than the average number of neighbors, we actually simulated a 550x550 meter network; however, we only collected statistics for nodes in the 350x350 meter center area. In other words, we removed the higher probability that perimeter nodes have for retransmitting in our simulation.

Figure 3.5 demonstrates that our mathematical model, given in Equation 3.10, compared to our simulation results, provides reasonably close estimates of the probability that a given node rebroadcasts. Our model, compared to our simulation results, slightly under predicts the probability of retransmission in sparse networks and slightly over predicts the probability of retransmission in dense networks. We conclude that the average number of neighbors (which is dependent on the total number of network nodes, transmission range, and network area) as well as the threshold value strongly affect the retransmission frequency of nodes using the Counter-Based scheme. We believe the results from our model do not exactly match the results from our simulation due to the assumptions we had to make in the development of our model. First, we assumed that each of v's neighbors received the packet at the same time as v; thus, v has a 50% chance of rebroadcasting before each neighbor. In our

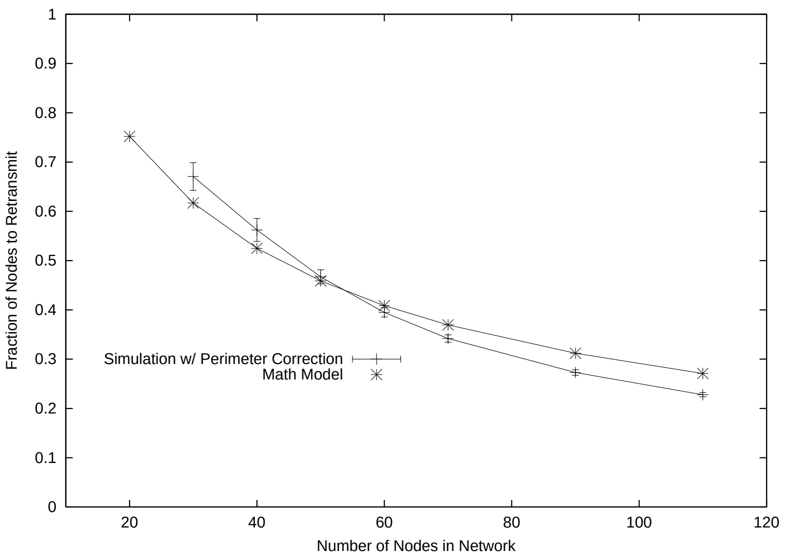

FIG. 3.5. Math Model Prediction of Retransmission Probability for the Counter-Based Scheme

simulation, we expect a packet to propagate to some neighbors before others which affects the overall retransmission frequency. Secondly, we assumed that each neighbor node has an independent probability (P) of rebroadcasting, which matches the overall probability of rebroadcasting in the network. In our simulation, we expect the probabilities of neighbors rebroadcasting to be dependent on one another and this dependency affects the overall retransmission probability.

3.3.2 Predicting P for the Distance-Based Scheme

Our mathematical model of P for the Distance-Based scheme is similar to the mathematical model P for the Counter-Based scheme. That is, we sum the individual probabilities for each case that would induce a node to rebroadcast a packet.

In the Distance-Based scheme, a node, v, will not rebroadcast if it receives a packet from a source node, u, that is within a threshold distance, D. Consider an annulus area centered at v, with the inside radius equal to D and the outside radius equal to the transmission distance, r, of the node's radio. If node u is within the annulus area (i.e., outside the circle with radius D and inside the circle with radius r), then v will start a RAD and wait for redundant packets. During the RAD, if any redundant packet is received from a node within D distance to v, then v will not rebroadcast.

To determine P, we start by determining scenarios that would induce node v to rebroadcast. For the Distance-Based scheme, the only scenario we need to consider is:

- Node v receives packets only from neighbors within the annulus area before node v's RAD expires.

Consider the case in which M neighbors of v rebroadcast the packet. (We note that M does not include the source node, u, for reasons we explain below.) Given M, we need to determine the probability that B nodes, where B is the number of v's neighbors that broadcast before v's RAD expires, are within the annulus area previously discussed. Since RAD times are chosen from a uniform and random distribution, and since we assume that each of v's neighbors receives the packet at the same time as v, then we expect 50% of the M nodes to rebroadcast before v.

Of the nodes that broadcast before v's RAD expires, we need to know the probability that all are inside the annulus area. The area of the annulus is $A_r - A_D$, where A_r is the transmission area of a node and A_D is the area of a circle with radius equal to D. Thus, the probability, P_a, that a neighbor of v is within the annulus is given by

$$P_a = \frac{A_r - A_D}{A_r}$$

or

$$P_a = 1 - \frac{D^2}{r^2} \ .$$

Therefore, the conditional probability of B given M is:

$$P[B = b \mid M = m] \ = \ \binom{m}{b} (0.5)^b \, (1 - 0.5)^{m-b} * (P_a)^b \, .$$

It is possible that all M rebroadcasting nodes rebroadcast before v. To account for each possibility, we sum the probabilities of the individual events. That is,

$$P[B \leq m \mid M = m] \ = \ \sum_{b=0}^{m} \left(\binom{m}{b} (0.5)^m * (P_a)^b \right) . \tag{3.11}$$

We must also account for the fact that the source node, u, must be within the annulus for v to rebroadcast. Thus, Equation 3.11 becomes:

$$P[B \leq m \mid M = m] \ = \ \sum_{b=0}^{m} \left(\binom{m}{b} (0.5)^m * (P_a)^{b+1} \right) . \tag{3.12}$$

Equation 3.12 provides the probability that node v will rebroadcast given that M of node v's neighbors rebroadcast the packet. Now we must determine the probability distribution of M. We assume that each neighbor rebroadcasts with a probability P. (We note that this is precisely the quantity that we seek to compute.) The probability that M neighbors rebroadcast is dependent on v's number of neighbors, I. Assuming a binomial distribution, we represent the conditional probability of event M given I

as:

$$P[M = m \mid I = i] \;=\; \binom{i}{m} (P)^m (1 - P)^{i-m} .$$

As was the case for B, it is possible for M to be a number of different values. Specifically, there is a possibility that M is any value value between 0 and i. To account for each possibility, we sum each probability:

$$P[M \leq i \mid I = i] \;=\; \sum_{m=0}^{i} \binom{i}{m} (P)^m (1 - P)^{i-m} . \tag{3.13}$$

Finally, we must determine the probability of I. Consider a node w that is a network node, but is not v nor the source node u. The probability that w is a neighbor of node v is P_n, as described in Equation 3.4. Since nodes are uniformly and randomly placed in the network, and since we assume the network has a total of N nodes, the probability that node v has I neighbors is:

$$P[I = i] \;=\; \binom{N-2}{i} (P_n)^i (1 - P_n)^{N-2-i} .$$

Again, there is a non-zero probability that I may be equal to any value between 0 and $N - 2$, inclusive. Thus, we sum each probability with:

$$P[I \leq N - 2] \;=\; \sum_{i=0}^{N-2} \binom{N-2}{i} (P_n)^i (1 - P_n)^{N-2-i} . \tag{3.14}$$

We noted earlier that the source node, u, is not accounted for in B. Nor is it accounted for in M or I. We note that u, by definition, is a neighbor of v and u is a rebroadcasting node and it rebroadcasts before v. The probabilities of these events is equal to 1.0. Therefore, u is removed from the probability calculations for the individual events described in Equations 3.12, 3.13, and 3.14.

To solve for P, we need to sum the probabilities of B given M given I. Thus, P is defined as:

$$
\begin{aligned}
P = \sum_{i=0}^{N-2} \Bigg[\binom{N-2}{i} (P_n)^i (1-P_n)^{N-2-i} \\
* \sum_{m=0}^{i} \Bigg(\binom{i}{m} (P)^m (1-P)^{i-m} \\
* \sum_{b=0}^{m} \binom{m}{b} (0.5)^m * (P_a)^{b+1} \Bigg) \Bigg].
\end{aligned}
\tag{3.15}
$$

We used a root finding function in Matlab to find the value of P satisfying Equation 3.15. We compared the results of our mathematical model with a simulation of the Distance-Based scheme. We used the same simulation conditions discussed in Section 3.3.1 for the Counter-Based scheme. Figure 3.6 illustrates that our model, compared to our simulation results, reasonably predicts retransmission frequency. We conclude that the average number of neighbors and the threshold distance value strongly affect retransmission frequency. This model is a better predictor for all node densities than the model for the Counter-Based scheme. Recall that two of the three scenarios in which a node rebroadcasts in the Counter-Based scheme involve assumptions that each neighbor broadcasts with the same probability P. For the Distance-Based scheme only one scenario is affected by this assumption. Since, in our simulation, we expect the probabilities of neighbors rebroadcasting to be dependent on one another, our assumption affects, to a higher degree, the more complicated model of the Counter-Based scheme.

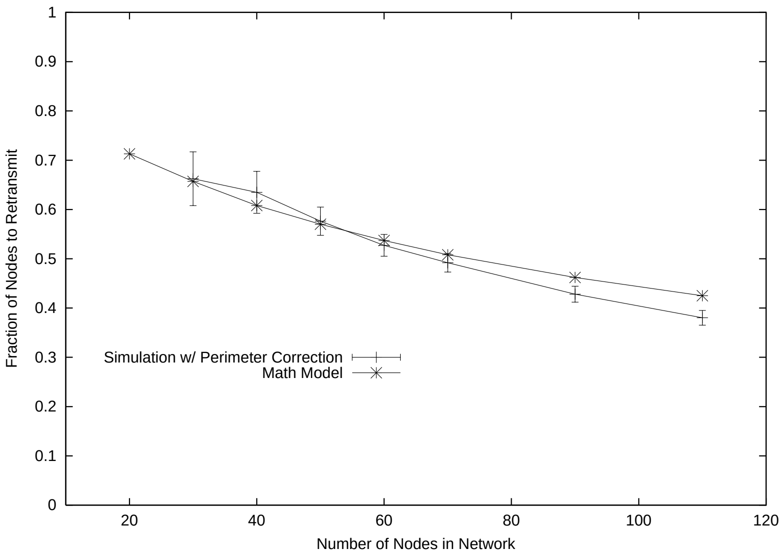

Fig. 3.6. Math Model Prediction of Retransmission Probability for the Distance-Based Scheme

Chapter 4

SIMULATION COMPARISON OF BROADCAST PROTOCOLS

4.1 Selected Protocols

As discussed in Section 1.2, only a subset of the broadcast protocols were implemented for simulation. We chose to implement Simple Flooding, the Counter-Based scheme, the Location-Based scheme, SBA and AHBP. This section explains our choices. Since only one protocol exists in the Simple Flooding category, this protocol is evaluated.

For the Probability Based methods, the choice was clear. Both the Probabilistic scheme and Counter-Based scheme were proposed in the same paper. Comparisons in [34] indicated that the Counter-Based scheme outperformed the Probabilistic scheme in most simulated conditions.

For a similar reason, the Location-Based scheme was chosen to represent the Area Based methods. In general, the Location-Based scheme was shown to be more robust than the Distance-Based scheme [34]. Our implementation mimics the original proposal by using a convex polygon approximation to determine the expected additional transmission coverage when the number of redundant receptions is greater than three. The authors of [34] were vague in their description of how the protocol reacts when the number of redundant receptions is less than three. Our version verifies that the source of each redundant transmission is farther than a threshold distance from the receiving node; otherwise the node does not rebroadcast. In other words, our simulated protocol combines the Distance-Based and Location-Based schemes.

Neighbor Knowledge is the biggest category of broadcast protocols, and contains seven protocols from six articles. We chose to implement two protocols to represent this category: SBA and AHBP. The following discussion justifies this choice.

The seven neighbor knowledge protocols can be classified by whether a node makes a local decision to retransmit a broadcast packet. A node that uses either Flooding with Self Pruning, SBA, or LENWB makes these local decisions. A node that uses Dominant Pruning, Multipoint Relaying, AHBP, or CDS-Based Broadcasting is told (either via the packet of via a previously sent control packet) whether it needs to retransmit a broadcast packet.

We chose SBA to represent the protocols that make local decisions on whether to rebroadcast. Although Flooding with Self Pruning was applicable in some network conditions, it was shown to be inefficient in others [35]. Similarly, the authors of LENWB [42] show that the base protocol performs poorly over a range of network conditions common to MANETs. SBA, on the other hand, uses neighbor knowledge efficiently and was shown to obtain good simulation results in [36].

We chose AHBP to represent neighbor knowledge protocols that do *not* make a local decision on whether to rebroadcast. AHBP uses a more efficient algorithm for selecting next hop rebroadcasting nodes than Dominant Pruning. In addition, AHBP would appear to benefit from the three ways in which it differs from Multipoint Relaying (as mentioned in Section 2.2.4). Other work on connected dominating set based routing in ad hoc networks include [44] and [45], both of which propose distributed algorithms to efficiently approximate a minimum CDS in a network. However, since AHBP and its mobility extension (AHBP-EX) are well defined in [40] we chose it over the CDS-Based broadcasting scheme.

4.2 Description of Studies

We provide a side by side comparison of the broadcast protocol categories in this thesis. Specifically our goals are to:

1. Compare the protocols over a range of network conditions including node densities, node mobility and traffic rates.

2. Pinpoint areas where each protocol performs well and areas where they could be improved.

3. Propose protocol enhancements to improve performance.

4. Discover if any one protocol stands out in its current form as being appropriate for diverse network conditions. That protocol may serve as an appropriate model for standardization or as a benchmark for further work.

To attain these goals we focus on a four part study. Each study is outlined in a subsection below. While our four studies vary some network parameters, the ones outlined in Table 4.1 remain constant for all simulations. Our network area and node transmission distances were chosen to allow reasonably timed simulation of networks characterized by high node density, congestion level and node mobility.

4.2.1 Study 1 - Algorithm Efficiency

The purpose of this study was to evaluate the core algorithms of the different protocols by comparing performance in a static network using a Null MAC. Each algorithm has a different level of complexity and one would hope that the more complex protocols have the best performance. The algorithms are highly dependent on the density of the network. In sparse networks, the protocols are expected to

Simulation Parameter	**Value**
Simulator	NS-2 (1b7a)
Network Area	350 x 350 meter
Node Tx Distance	100 meter
Data Packet Size	64 bytes payload
Node Max. IFQ Length	50
Simulation Time	100 seconds
# of Trials	10
Confidence Interval	95%

Table 4.1. Simulation Parameters Common to All Studies

perform similar to Flooding, as each node may have to rebroadcast to reach isolated neighbors. As density increases, proportionally fewer nodes should rebroadcast.

In addition to the "worst-case" bound provided by Simple Flooding, we also include a theoretical "best-case" bound provided by the Minimum Connected Dominating Set (MCDS). An MCDS is the smallest set of rebroadcasting nodes such that the set of nodes are connected and all non-set nodes are within one-hop of at least one member of the MCDS. It is impossible for any algorithm to perform better than the MCDS and unlikely to perform worse than Simple Flooding. Thus, these two bounds provide a useful spectrum to gauge the relative performance of the protocols. Most previous broadcast papers compare specific protocols to Flooding; to our knowledge, this "best-case" bound comparison has not been done in any previous work.

We note that the determination of an MCDS is an NP Hard problem. A number of papers have proposed *approximation* algorithms to determine MCDS, e.g., [46]. However, since we want the exact MCDS, we implemented a Brute Force and Ignorance (BFI) method that, given a static topology, iterates through every possible node combination for a proposed MCDS size to determine if indeed there is an MCDS

# Network Nodes	Avg # of Neighbors
20	3.8
30	6.0
40	7.6
50	9.1
60	11.2
70	13.9
90	18.1
110	21.2

Table 4.2. Average Number of Neighbors for Different Numbers of Network Nodes

of that size.

For this study, we varied the number of nodes randomly placed in the network area from 20 to 110. Table 4.2 shows average number of neighbors (i.e., the actual neighbor count for each node averaged over 10 runs). Our average number of neighbors covers the gamut of sparse to dense networks.

By using a static network and Null MAC, we avoid any effects that mobility and congestion may have on the protocols. In other words these conditions give a good quasi-theoretical view of the core algorithms in the protocols. In addition these conditions also approximate networks which may be characterized by high stability and very low traffic rates. A broadcast packet origination rate of 10 packets per second was used, although the use of a Null MAC renders the origination rate irrelevant.

4.2.2 Study 2 - Congested Networks

The purpose of this second study is to quantify the effect of congestion on each of the protocols. Obviously the delivery ratio of Flooding will be compromised in this study, as a congested network causes numerous collisions and/or queue overflows.

The other protocols, as more efficient schemes, should be less sensitive to congestion. This study was performed using the contention based 802.11 MAC scheme.

Congestion can be obtained by increasing packet size, increasing frequency of packet origination or both. We chose to fix the packet size and vary the packet origination rate (source rate) because we anticipate broadcast packets, as control type packets, to generally be small in size. In this study, the payload portion of each packet was set at 64 bytes and the rate was varied from 1 packet per second to 80 packets per second. To create a random traffic pattern, each new packet was assigned a source node randomly chosen from the entire pool of network nodes.

The number of network nodes was set at 60, which roughly represents the median value from Study 1. One might expect significantly different results using different numbers of network nodes. However, the goal of this study is to obtain a general trend for the effect of a congested network.

A static network was used for this study to ensure that the effects of mobility would not interfere with the effects of congestion. Any anomalies in one static network topology is accounted for by averaging the results over 10 unique topologies for each packet source rate.

4.2.3 Study 3 - Mobile Networks

This study focuses on the ability of each protocol to react effectively to node mobility in the network. Previous work has demonstrated that Flooding is relatively insensitive to mobility [32]; the optimized schemes should mimic that behavior.

A Null MAC is used in this study to ensure no effect from congestion. The packet source rate was set at 10 packets per second; since all results are provided on a per packet originated basis, and a Null MAC is evaluated, the packet source rate is fairly inconsequential. Similar to Study 2, the number of network nodes was set at 60.

We use the random waypoint mobility model [47] with zero pause time. The range of mean speeds is varied from 1 to 20 meters per second in our simulations. Nodes chose a random location in the simulation area, and moved to that location at a given mean speed plus or minus 10%. When a node reached the location, it immediately chose a new random location and speed and moved to that new location at the newly chosen speed.

The choice to use zero pause time and this range of speeds is somewhat arbitrary. Previous work shows that pause time adds significant stability to dynamic networks; [48] shows that given a network of nodes with mean speeds of 10 m/s, pause times of 50 seconds can reduce the number of link changes by 50%. Since we preferred to test the protocols without this added stability, we chose to use zero pause time. One could argue that our simulations are unrealistic especially at the 20 meter per second scenario. We agree that it is unlikely to have nodes moving at close to 50 miles per hour in this pattern in a 350x350 meter space. However, we view mobility as a proxy for link change rates in general. For example, links may break from nodes moving behind obstructions (or vice versa), from interference from other utilizations of the public bandwidth, or even from node devices being turned off. Although we have no supportive data to conclude that these mobile scenarios match real world networks in terms of link change rates, the range of node speeds does provide a diverse range of node behavior.

4.2.4 Study 4 - Combined Networks

In the previous three studies we focus on particular network conditions by varying node density, congestion and network mobility separately. In order to isolate the

Trial	1	2	3	4	5
# of Nodes	40	50	60	70	90
Avg. Speed (m/sec)	1	5	10	15	20
Pkt. Src. Rate (pkts/sec)	10	20	40	60	80

Table 4.3. Study 4 Trial Simulation Parameters

effects of those changes, we only varied one parameter. The downside is the behavior at one set of constant parameters may be different than the behavior at another set of constant parameters. Also, we risk missing the combined effects of mobility, congestion and node density.

This study attempts to address those concerns. To perform a complete study, one should simulate all combinations of node density, node speed and packet source rates. Since that breadth of coverage isn't appropriate for the scope of this thesis, we chose to aggregate parameters into five trials to give an overview of performance at combined conditions.

In the previous studies we increase node density, traffic source rates and average node speed. In other words, we evaluate the protocols as the severity of the network environment increases. We maintain this trend of increasing severity along the X-axis in this study. The five trials are designed so that Trial 1 takes a combination of the least severe conditions and Trial 5 takes a combination of the most severe conditions. The specific parameters are shown in Table 4.3.

This particular aggregation of the three previous studies provides several bits of important information. Among others things, it demonstrates how the protocols react in real networks. It allows us to gauge relative importance of node density, mobility, and congestion to the overall performance of each protocol. It illustrates the general limits of each protocol for a given network environment. Lastly, it provides a cursory indication of which protocol reacts best over a range of network severities.

4.3 Simulation Results

In addition to the simulation parameters provided in Section 4.2, protocol parameters must be specified. Important protocol parameters include threshold values used by the Counter-Based and Location-Based schemes in the decision on whether to rebroadcast. In addition, RAD times affect all protocols except Flooding and AHBP. Finally, SBA and AHBP both use "Hello" packets, thus a beaconing period is needed as well. Our protocol parameter choices for each study are discussed in each study subsection.

As mentioned in Section 4.2, each point on each graph presented in this section is the average of 10 simulation runs. We include error bars which indicate 95% confidence that the actual mean is within the range of said interval. In certain cases, the confidence intervals are small enough that they are obscured by the symbol itself.

4.3.1 Study 1 - Algorithm Efficiency

Protocol parameters for all simulations in Study 1 and Study 2, unless otherwise noted, are shown in Table 4.4. The Counter-Based and Location-Based threshold values were chosen to minimize the number of retransmitting nodes with the constraint of maintaining at least 95% delivery ratio in a 30 node network (the effects of threshold values are illustrated later in this section). RAD T_{max} times are somewhat arbitrary given the use of the Null MAC; their effect is addressed in Study 2. While SBA and AHBP perform well in this study, they do incur an additional cost associated with the use of "Hello" packets. However, the network in this study is static;

Parameter	Value
CB Threshold	3
LB Threshold	45 meters
RAD T_{max}	0.01 seconds
“Hello” Beacon Interval	N/A

Table 4.4. Protocol Parameters used for Study 1 and Study 2

thus, the “Hello” packet beaconing interval is inconsequential as neighbor sets do not change over the course of the simulation. Therefore, the “Hello” packet cost is not considered in this first study.

Figure 4.1 shows delivery ratio (the percent of network nodes who receive any copy of a broadcast packet) for each protocol as the number of nodes is increased in a static network with a Null MAC. When 30 nodes or higher exist in the network, over 95% delivery ratio is acheived for all protocols. Setting the threshold values for the Counter-Based and Location-Based schemes to levels that ensured over 95% delivery ratio with 20 network nodes seemed overly constraining and resulted in excessively high numbers of rebroadcasting nodes for those protocols at higher node counts.

Figure 4.2 shows the number of retransmitting nodes for each protocol as the node density increases. The figure illustrates that each broadcasting scheme except Flooding is scalable in terms of higher node density in a fixed network area. As expected, the protocols utilizing more complex algorithms require the least number of rebroadcasts. In general, the neighbor knowledge schemes require fewer rebroadcasts than the area based schemes, which require fewer broadcasts than the probability based schemes. In other words, neighbor knowledge schemes benefit from the “Hello” packet and high algorithmic costs by having fewer nodes retransmit each broadcast packet. In sparse networks all schemes converge to Flooding, as each node must

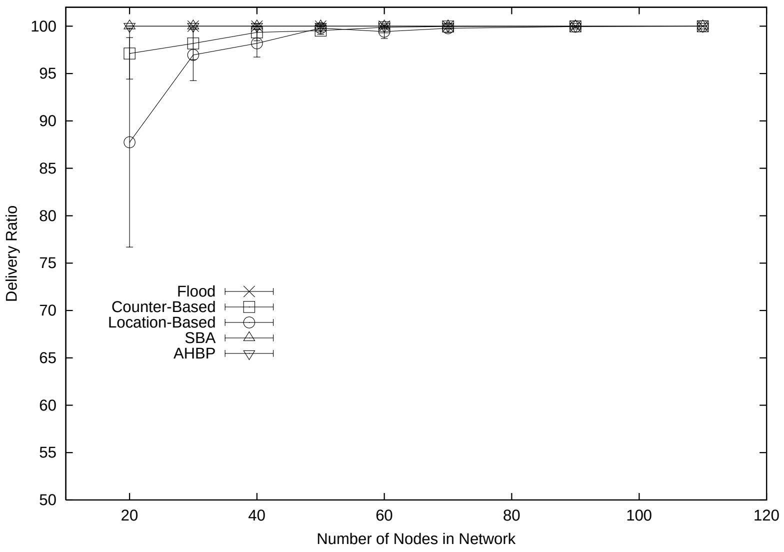

FIG. 4.1. Delivery Ratio versus Number of Network Nodes

retransmit each packet. In dense networks (e.g., in the simulations with 110 nodes), the theoretical best case MCDS requires 8% of the nodes to rebroadcast versus AHBP which requires 18% of the network nodes to rebroadcast. While there is room for improvement, AHBP approximates the minimum fairly well. Note that actual MCDS values were only calculated for networks up to 70 nodes; the BFI calculations were too computationally intensive for networks with more than 70 nodes. We project MCDS for more than 70 nodes in the network with a dotted line.

We note that the MCDS size is relatively constant over the range of node densities in Figure 4.2, which intuitively makes sense. Given a fixed node transmission distance, MCDS size should only increase with network area. Figure 4.3 illustrates the effect of changing network size while the number of nodes stays constant and transmission range stays constant. The X-axis is the ratio of Network Area to Transmission Area. MCDS size varies directly with this ratio.

In the previous two graphs, the threshold values for the Counter Based and Lo-

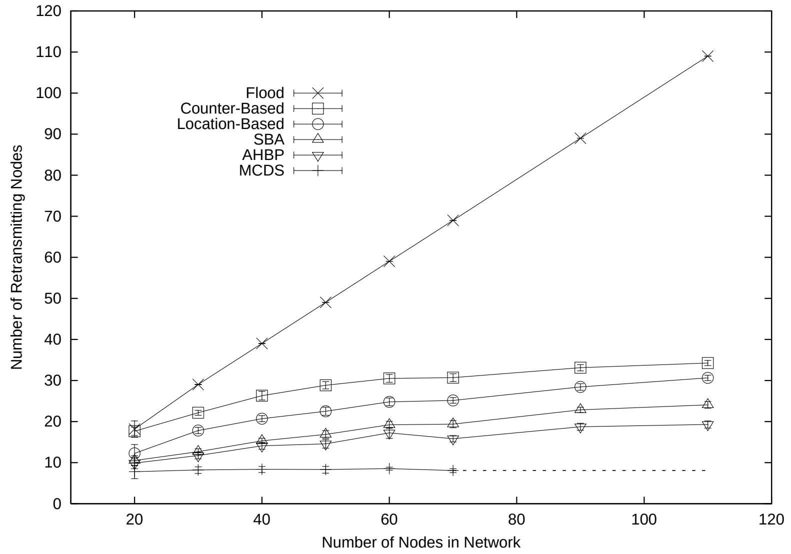

Fig. 4.2. Number of Retransmitting Nodes versus Number of Network Nodes

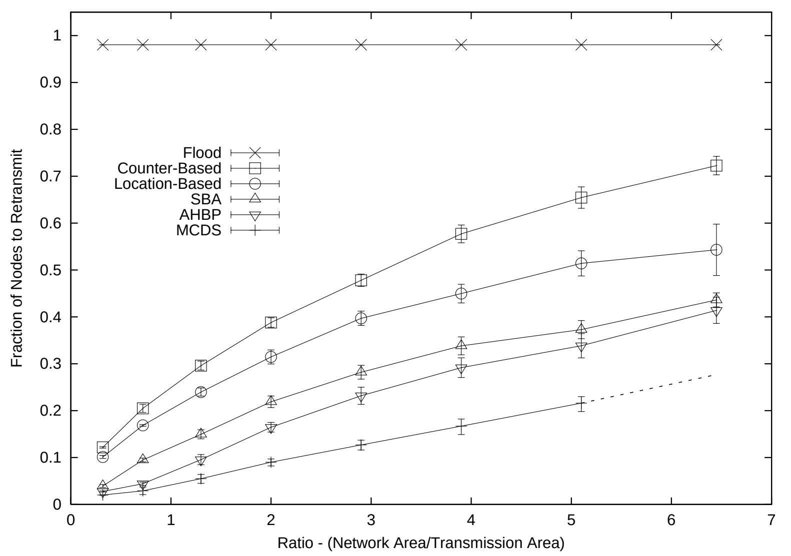

Fig. 4.3. Fraction of Retransmitting Nodes versus Area Ratio

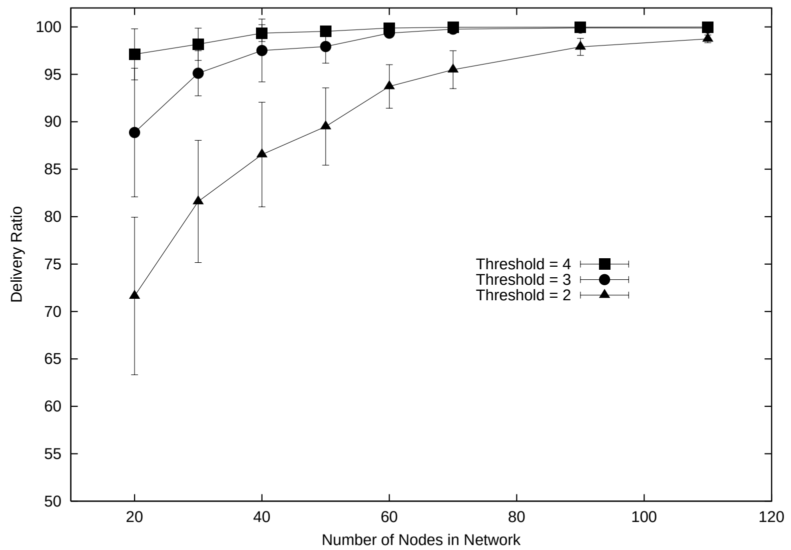

FIG. 4.4. Counter Based Scheme - Sensitivity of Delivery Ratio to Threshold Value

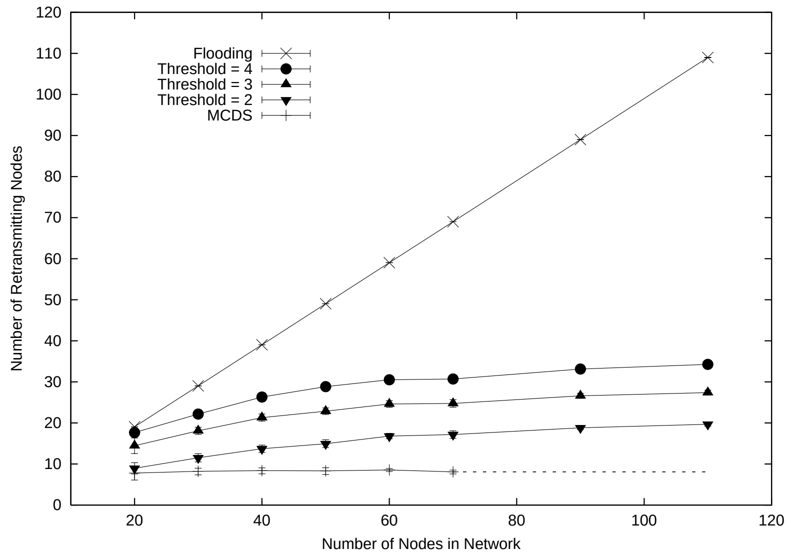

FIG. 4.5. Counter Based Scheme - Sensitivity of Number of Retransmitting Nodes to Threshold Value

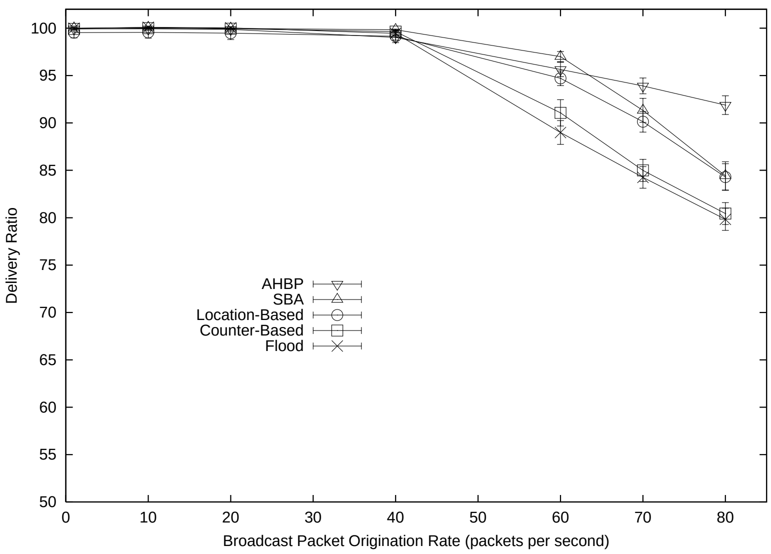

FIG. 4.6. Delivery Ratio versus Packet Origination Rate

cation Based Schemes were held constant. Those values, however, are not optimal over the range of node densities. Figure 4.4 shows the delivery ratios for the Counter Based scheme given various threshold values. To maintain a high delivery ratio in sparse networks, a higher threshold is needed; for more dense networks, the counter based scheme can maintain a high delivery ratio with a lower threshold. Figure 4.5 illustrates that a lower threshold value means fewer nodes retransmit. Thus, extending the Counter-Based scheme to adapt its threshold value based on neighbor density would improve the overall performance of the protocol. This extension would likely require the use of “Hello” packets. One could argue, however, that if “Hello” packets are used, a protocol should use the available neighbor information more intelligently such as SBA or AHBP.

A similar study was performed for the Location-Based scheme in which threshold values were evaluated at 35, 45, 65 and 85 meters. Results show that a threshold value of 35 meters is preferred for sparse networks and 85 meters is preferred for

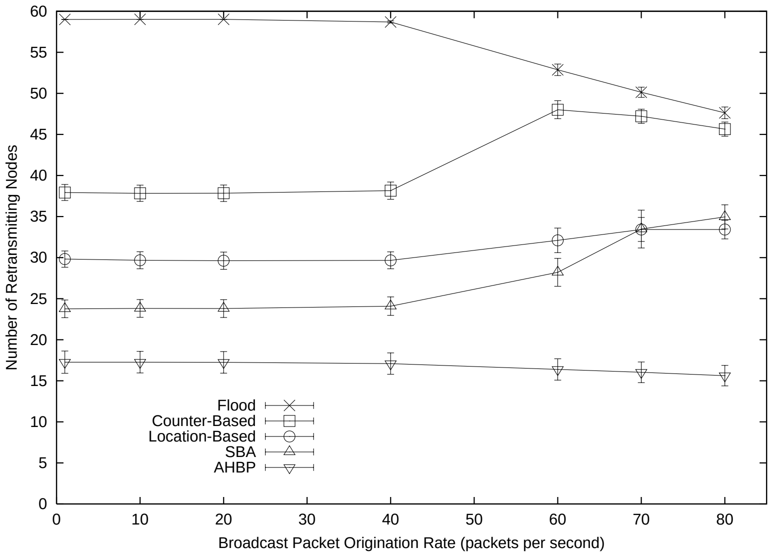

FIG. 4.7. Number of Retransmitting Nodes versus Packet Origination Rate

dense networks. Thus, the Location-Based scheme would also benefit from adaptation based on neighbor density.

4.3.2 Study 2 - Congested Network

Recall that the goal of Study 2 is to test the protocols under the constraint of limited bandwidth using the IEEE 802.11 MAC. Thus, all figures presented in this section increase the number of broadcast packets originated per second along the X-axis.

All simulations in this section were performed in a static network. The protocol parameters used in this study are given in Table 4.4. As mentioned in Study 1, the use of a static network makes the "Hello" packet beaconing rate inconsequential.

Figure 4.6 illustrates that each protocol suffers as the network becomes more congested. Comparing this figure to Figure 4.2 shows the relationship between performance in congested networks and the number of redundant retransmissions: proto-

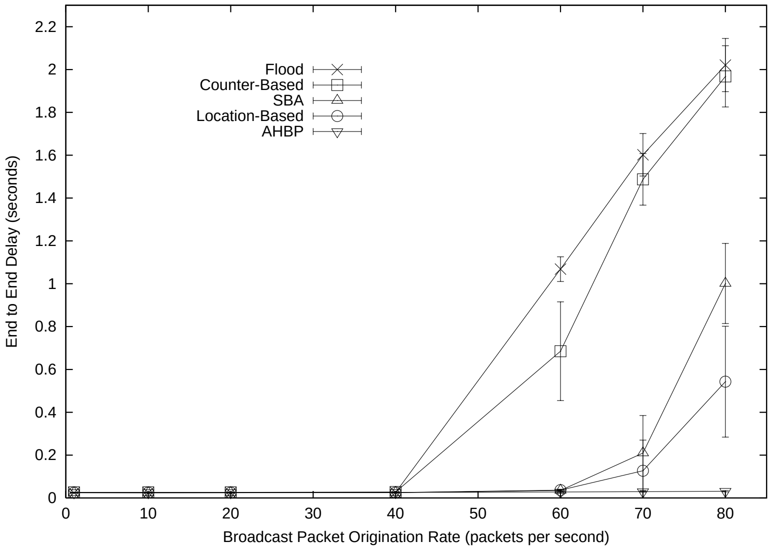

FIG. 4.8. End to End Delay versus Packet Origination Rate

cols that minimize the number of redundant retransmissions deliver the most packets in congested networks.

Figure 4.7 shows the number of retransmitting nodes as the network becomes more congested. Since the number of network nodes and the simulation area remain constant, one might expect the number of retransmitting nodes to remain constant in Figure 4.7 as well. Indeed, AHBP seems to basically follow this trend. The Counter Based, Location Based and SBA protocols, on the other hand, incur increased number of retransmissions per broadcast packet as the number of broadcast packets originated increase. This increase in retransmissions is due to the RAD utilized by each of these three protocols. In the Null MAC case of Study 1, the RAD had little consequence because delivery of packets was instantaneous. In this study, the delivery of packets is subject to delay due to the transmission times, back-offs from failed clear channel assessments, and blocked IFQs. Essentially, higher congestion prohibits redundant packets to be delivered during the RAD; therefore, more nodes rebroadcast. More

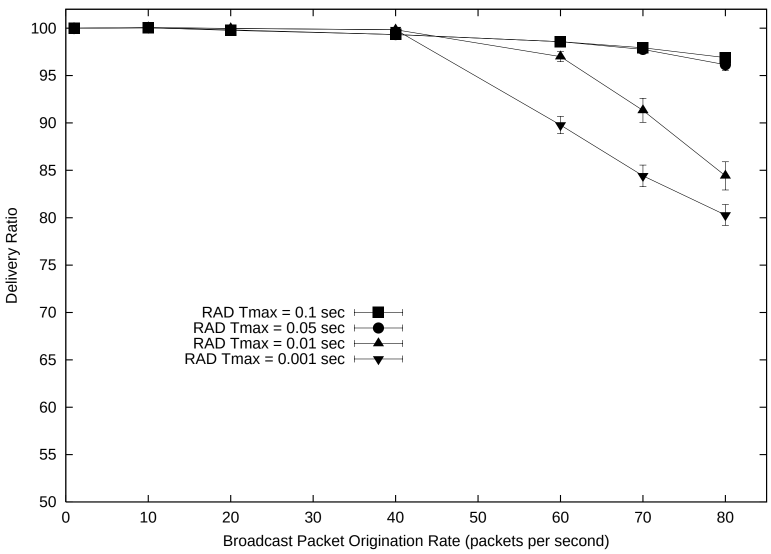

FIG. 4.9. Scalable Broadcast Algorithm - Sensitivity of Delivery Ratio to Random Assessment Delay T_{max}

rebroadcasts further congest the network resulting in a snowball effect. For Flooding, the number of retransmitting nodes drops as the network becomes congested, which directly illustrates the effect of collisions and queue overflows in congested networks.

Figure 4.8 verifies that there is a strong correlation between end to end delay (the time it takes for the last node to receive a given packet) and the origination rate of packets in the network. To maintain high delivery rates and low end to end delay, we propose that RAD adapt to the state of network congestion. We evaluate this proposal in the following discussion.

Figure 4.9 shows the delivery ratio and Figure 4.10 shows the end to end delay of SBA for different T_{max} values used in RAD determination. Remember, the RAD time is randomly chosen between 0 and T_{max} seconds. Note that the higher assessment delay is effective in increasing the delivery ratio of SBA in congested networks. Combining the results in Figures 4.9 and 4.10 provides the following two conclusions.

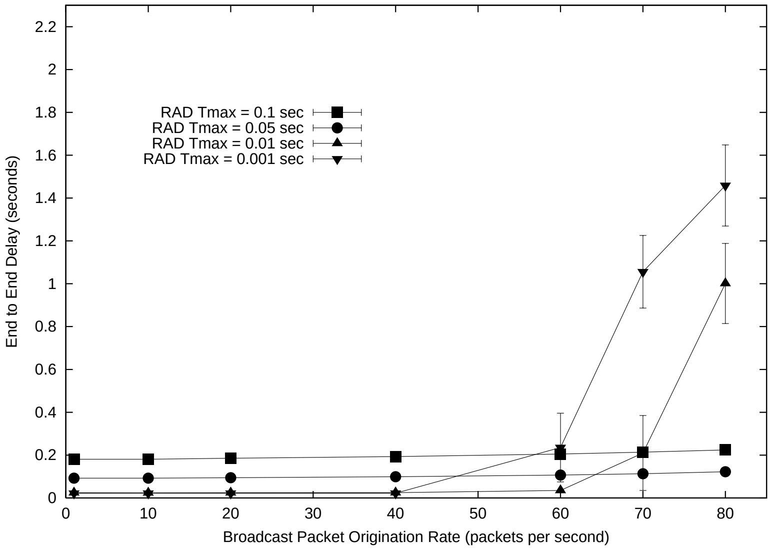

FIG. 4.10. Scalable Broadcast Algorithm - Sensitivity of End to End Delay to Random Assessment Delay T_{max}

First, when a network is not congested, low RAD values are desired. Second, when a network is congested, high RAD values are desired. Adapting RAD in this manner maximizes delivery ratio and minimizes end to end delay. The rest of this section describes results for our adaption of SBA to congestion. A similar adaptation is possible for any protocol that uses a RAD (e.g., the Counter-Based and Location-Based schemes); we note however that the adaptation lends itself more directly to protocols already using "Hello" packets.

In Section 2.1.2, we show that queue lengths are not a good indication of congestion in the network in our simulations. Therefore, to adapt SBA's RAD T_{max} to congestion levels, each node keeps track of the number of packets received per second. Table 4.5 provides average packet reception rates for SBA given various broadcast packet origination rates. A clear relationship exists between congestion and number of packets received. Our implementation only measures packets received because our

Pkt Orig. Rate	Avg. Rcvd. Pkts / Second
10	49
40	192
60	324
80	407

Table 4.5. Average Packet Reception Rate for a Given Packet Origination Rate

simulations only include SBA broadcast packets of the same size. Tracking bytes received per second would be a better metric for congestion in networks utilizing a number of different packet types and sizes. Our SBA adaptive scheme is a simple one: if the node is receiving on average more than 260 packets per second (which correlates roughly to a broadcast packet origination rate of 50 packets per second), the node uses a RAD T_{max} time of 0.05 seconds. Otherwise, the node uses a RAD T_{max} time of 0.01 seconds.

Figure 4.11 shows the performance of the adaptive SBA scheme against the original SBA scheme and AHBP. The adaptive SBA is the best performer with respect to delivery ratio; in fact, it is the only protocol with 95% delivery ratio for all congestion levels. While not shown, the adaptive SBA does require more retransmitting nodes per sourced packet than AHBP (e.g., 38% of network nodes versus 28%); the better delivery ratio comes at a cost of individual node resources.

This study shows that methods that minimize the number of rebroadcasting nodes deliver the highest number of packets in congested networks. Thus, we conclude that (in general) neighbor knowledge methods out perform area based methods which out perform the probability based methods in congestive situations. The improved performance comes at higher algorithmic cost and for neighbor knowledge methods only, processing cost of “Hello” packets.

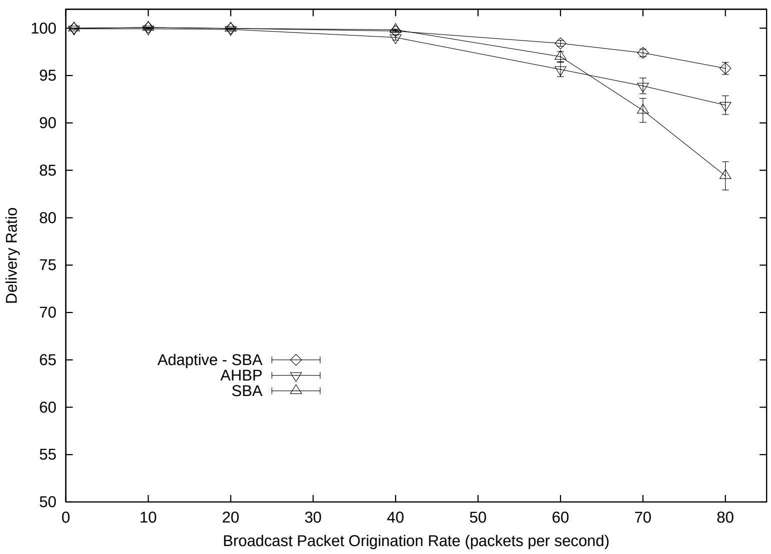

FIG. 4.11. Delivery Ratio for Adaptive Scalable Broadcast Algorithm

Parameter	**Value**
CB Threshold	3
LB Threshold	45 meters
RAD T_{max}	0.01 seconds
"Hello" Beacon Interval	1 per second

Table 4.6. Protocol Parameters used for Study 3 and Study 4

4.3.3 Study 3 - Mobile Network

This study focuses on the response of the protocols to mobile environments; we track performance as mean node speed increases along the X-axis. To remove any congestion effects, a Null MAC was used. (The use of a Null MAC precluded the need to simulate the Adaptive SBA defined in Study 2.) We chose 60 nodes (i.e., the median value from Study 1) and a 10 packet per second source rate. Protocol parameters for Study 3 are shown in Table 4.6.

Figure 4.12 shows that AHBP suffers from a changing topology. The mobility

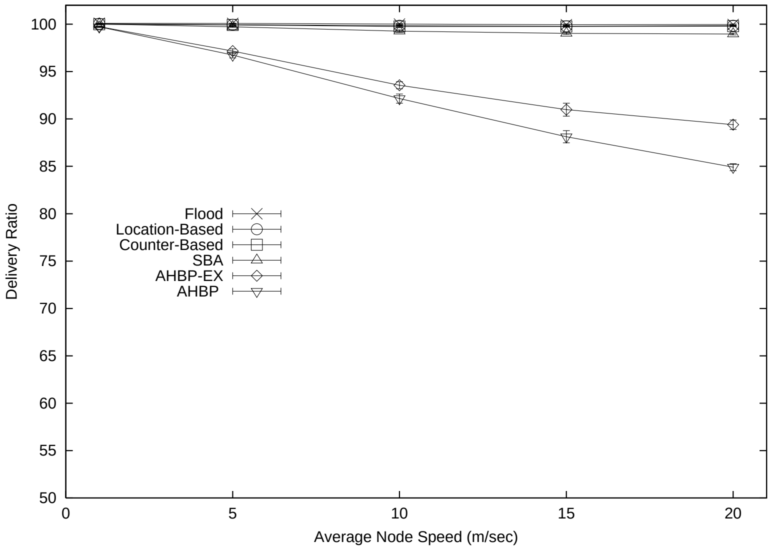

FIG. 4.12. Delivery Ratio versus Average Node Speed

extension for AHBP (AHBP-EX) improves AHBP; however, it still underperformed the other protocols. As discussed in Section 2.2.4, AHBP-EX requires that a node which receives a packet from an unrecorded-recorded neighbor (i.e., a neighbor not currently listed as a 1-hop neighbor) to act as a border relay gateway (BRG). In other words, AHBP-EX handles the case when a neighbor moves inside a node's transmission range between "Hello" intervals. The extension does not handle the case when a chosen BRG is no longer within the choosing node's transmission range. No recourse is provided in AHBP-EX to cover the 2-hop neighbors that this chosen BRG would have covered.

SBA, another neighbor knowledge method, is less sensitive to changing topologies than AHBP because SBA has each node assess its own topology. Recall that if a SBA node receives a packet from a new neighbor, it is unlikely to know of any common 1 or 2-hop neighbors previously reached; thus the node is more likely to rebroadcast. As shown in Figure 4.13, SBA naturally adapts to mobility by requiring more nodes to

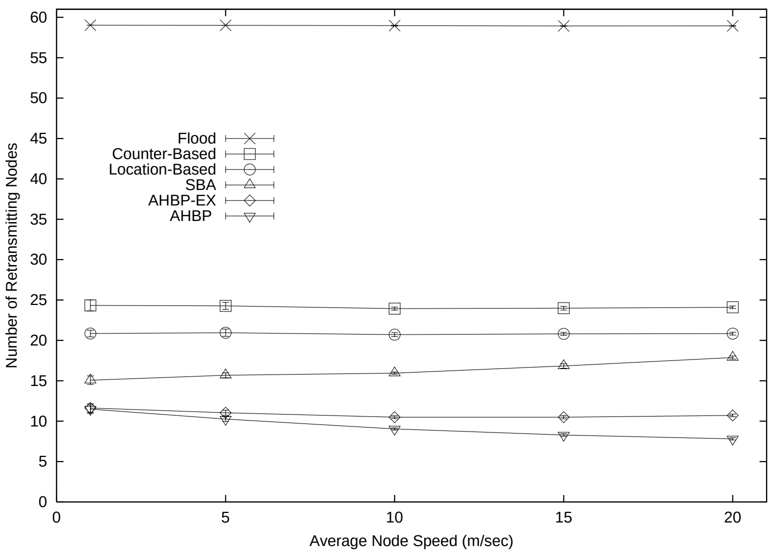

FIG. 4.13. Number of Retransmitting Nodes versus Average Node Speed

rebroadcast. AHBP and AHBP-EX, on the other hand, have fewer nodes rebroadcast as speed increases.

We evaluate the total transmission overhead (broadcast packets plus "Hello" packets) of AHBP and SBA and compare them to the protocols that do not require periodic beaconing in Figure 4.14. Figure 4.14 illustrates that at the one second beaconing period, AHBP-EX remains the least transmission expensive; SBA, on the other hand, becomes more expensive than the Location-Based Scheme and as expensive as the Counter-Based scheme at high mobility.

Neighbor knowledge methods must use "Hello" packets judiciously; otherwise the cost of beaconing outweighs the benefits of having fewer rebroadcasting nodes. As an example, one viable adaptation technique to allow AHBP-EX to correct for high mobility is to use shorter intervals between "Hello" packets. Figure 4.15 demonstrates that shortening the "Hello" interval from 1 second to 0.2 seconds allows AHBP-EX to maintain a high delivery ratio with a changing topology. Figure 4.16 shows the cost

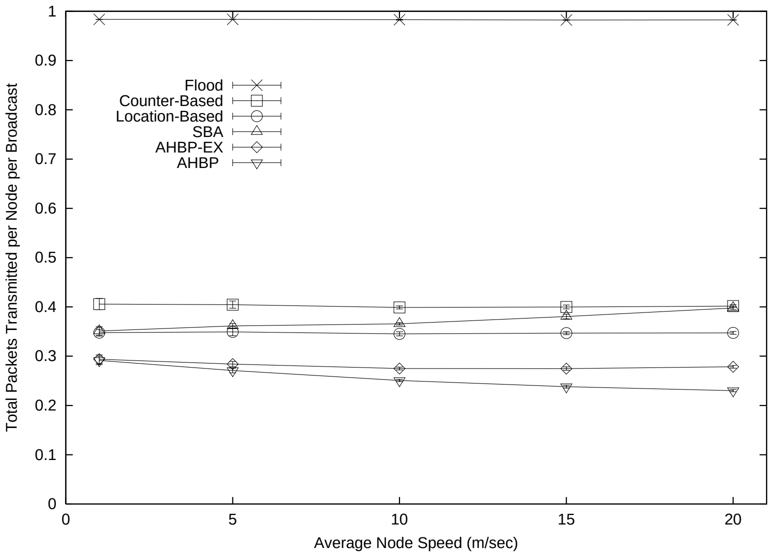

Fig. 4.14. Total Packets Transmitted per Node per Broadcast Originated versus Average Node Speed

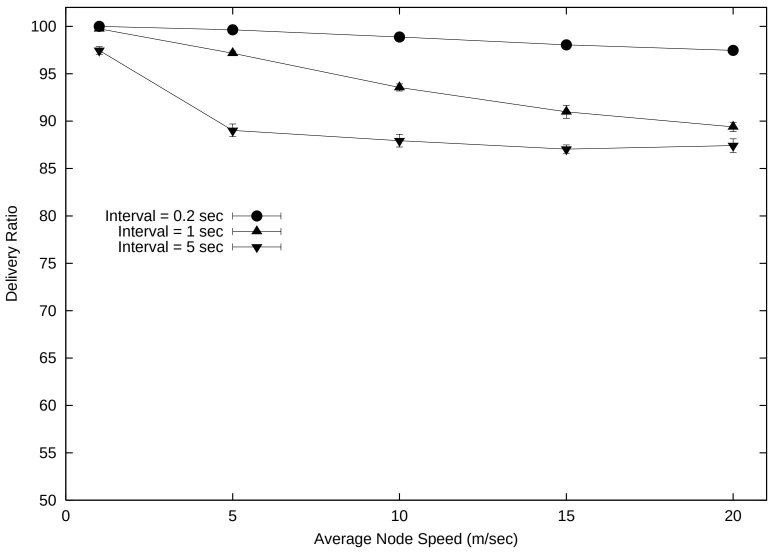

Fig. 4.15. Extended Ad Hoc Broadcast Protocol (AHBP-EX) - Sensitivity of Delivery Ratio to “Hello” Packet Interval

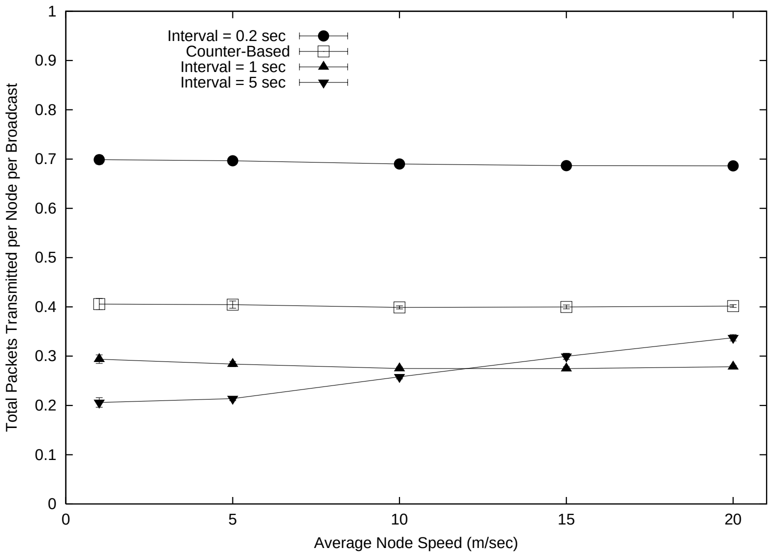

FIG. 4.16. Extended Ad Hoc Broadcast Protocol (AHBP-EX) - Sensitivity of Total Packets Transmitted to "Hello" Packet Interval - Compared to the Counter-Based Scheme

of the shorter interval in terms of total packets transmitted per node per broadcast and compares the cost to that incurred by the Counter-Based scheme. The shortest interval (0.2 seconds) is required for AHBP-EX to have a delivery ratio above 95%, but the total transmissions needed are nearly twice that required by the Counter-Based scheme. Furthermore, while a node only transmits one "Hello" packet per update interval, it will receive a "Hello" packet from each of its neighbors. The reception overhead is high, especially in dense networks.

We note that the broadcast traffic packet origination rate of 10 packets per second means the cost of the "Hello" packets was amortized over a moderately low number of broadcast packets. A higher traffic rate, while hiding the "Hello" packet cost in a larger number of total transmissions, still produces the same general trend: decreasing the "Hello" packet interval incurs node resource costs that outweigh the performance gains.

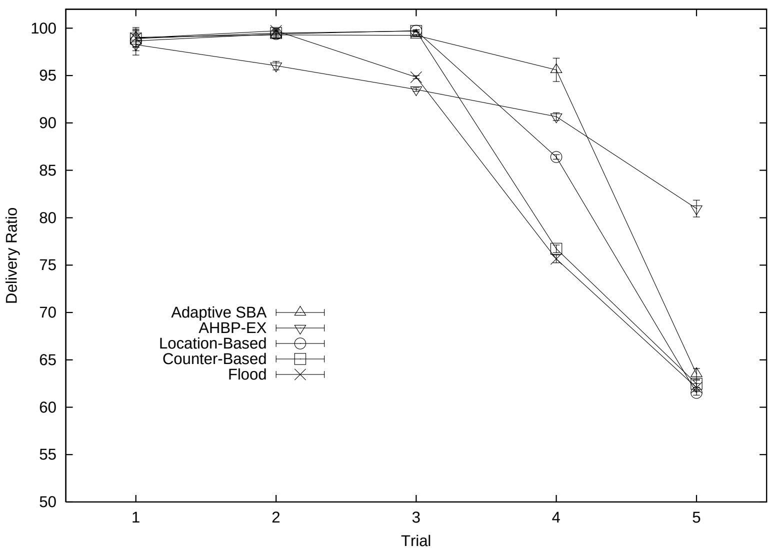

FIG. 4.17. Delivery Ratio as Severity of Network Environment Increases

4.3.4 Study 4 - Combined Network

In the previous three studies, we evaluated increasing the number of nodes, the amount of congestion and the degree of a changing topology independent of each other. In this study we evaluate the effect of all three parameters changing simultaneously by aggregating parameters into trials. Table 4.3 lists specific parameters; to summarize, as the trial number increases, the severity of the network environment also increases. Combining the conditions allows us to make observations regarding the relative effects the conditions have on each other. The protocol parameters used for this study are provided in Table 4.6.

Figure 4.17 shows delivery ratios for each protocol in each trial. As the severity of the network environment increases, each protocol has a "breaking point" in terms of its ability to deliver packets. Flooding breaks first, after Trial 2. The Counter-Based and Location-Based protocols break second, after Trial 3, and the Neighbor Knowledge protocols break last, after Trial 4. The Adaptive SBA scheme is the

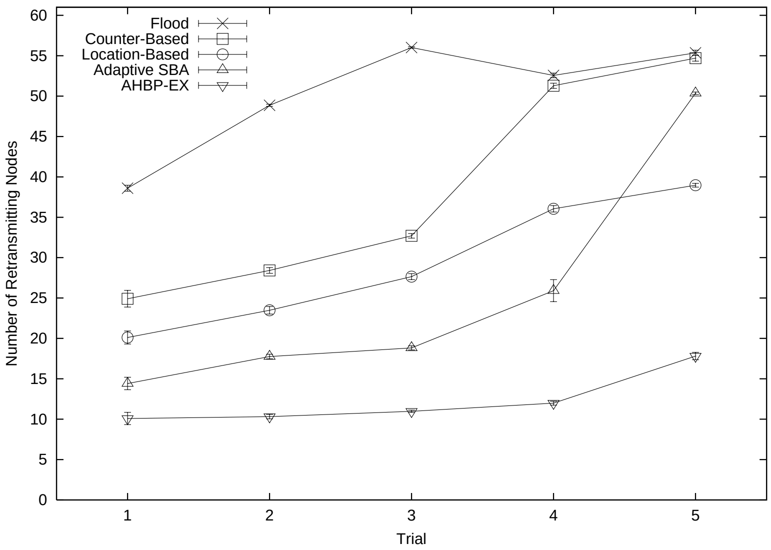

FIG. 4.18. Number of Rebroadcasting Nodes as Severity of Network Environment Increases

highest performer through Trial 4, after which congestion appears to catastrophically interfere with its ability to deliver packets. Our simple adaptation described in Study 2 was not sufficient to handle the congestion caused by high packet origination rate coupled with high node count. A more robust adaptation, which uses longer RADs, is possible and is an appropriate topic for future research.

The changing topology causes AHBP-EX to be the worst performer through the first three trials. However, AHBP-EX degrades the most gracefully, and in Trial 5 it outperforms all other protocols by roughly 20%.

Figure 4.18 shows the number of retransmitting nodes for each trial. All protocols experience an increase in the number of rebroadcasting nodes as the severity of the network environment increases. In fact, the Counter-Based scheme appears to mimic the behavior of Flooding after Trial 3 in terms of both the number of retransmitting nodes and the delivery ratio provided. As expected, the neighbor knowledge methods

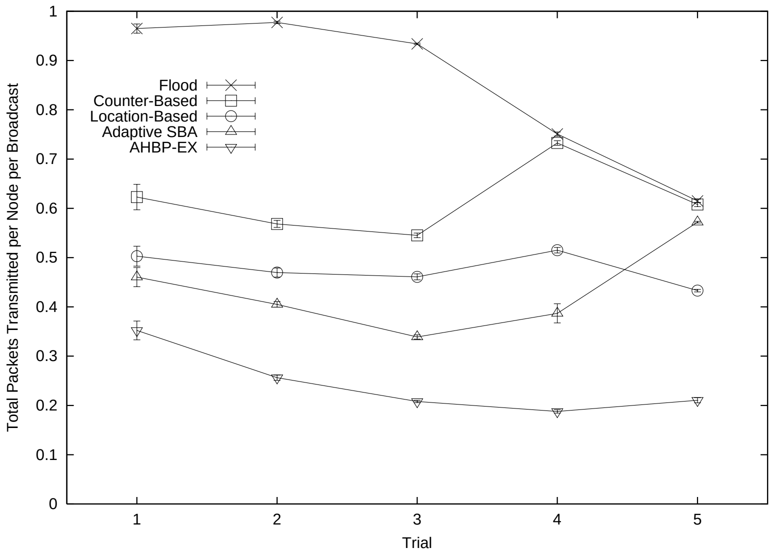

FIG. 4.19. Total Packets Transmitted as Severity of Network Environment Increases

have the fewest number of retransmitting nodes.

Figure 4.19 displays the total number of packets transmitted per node per broadcast packet originated. Comparing this figure to 4.18 illustrates the cost of "Hello" packets (at 1 per second) for SBA and AHBP-EX. At the low severity trials, the cost of amortization over broadcast packet source rate is high compared to that of the higher severity trials. In terms of total transmissions, AHBP and SBA offer the least cost compared to the other protocols (at least through Trial 4 for SBA).

Figure 4.20 shows end to end delay as network severity increases. The end to end delay results follow the trends shown in Figure 4.17.

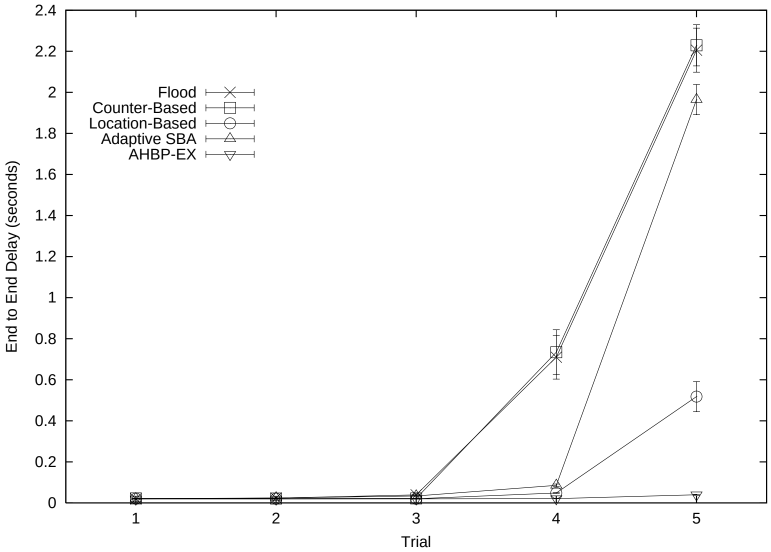

FIG. 4.20. End-End Delay as Severity of Network Environment Increases

Chapter 5

CONCLUSIONS

Proposed broadcasting schemes can be categorized into four families, Simple Flooding, Probability Based, Area Based and Neighbor Knowledge methods. The categories are listed in order of increasing algorithm complexity. The additional computation cost exists in order to reduce redundant transmissions while attempting to ensure that each node receives each broadcast packet.

Analysis of algorithm run time complexity demonstrates that each protocol incurs similar processing cost in terms of duplicate packet detection. Neighbor knowledge methods incur costs for both neighbor table maintenance and retransmission determination that is proportional to neighbor density. The Neighbor Knowledge methods incur the highest processing cost in all categories. The more complex Neighbor Knowledge methods are three orders of magnitude more complex than Simple Flooding.

We show that attempting to predict the fraction of network nodes that retransmit a given broadcast packet, based on the fraction of neighbor nodes versus network node density, is feasible. Our prediction is accomplished by assuming each node (except the source) has the same probability of retransmission. This allows us to determine the individual probabilities of a node retransmitting given specific topological conditions and event ordering. The summation of the individual probabilities provides the overall probability that a given node will retransmit a broadcast packet. The accuracy of our prediction model versus simulation is within 5%.

In order to compare broadcast protocols under realistic conditions of limited bandwidth and dynamic topologies, we compare a subset of the broadcasting protocols via simulation. We make the following conclusions from our performance evaluation of broadcast protocols:

1. Increasing node count in a static network disproportionately hurts the Probability Based and Area Based schemes in terms of number of rebroadcasting nodes. SBA and AHBP, two Neighbor Knowledge methods, approximate MCDS fairly closely as the number of nodes in the network is increased.

2. The schemes that utilize a RAD suffer in congestive networks unless a mechanism to adapt a node's RAD to its local congestion level is implemented.

3. The Neighbor Knowledge methods that do not use local information to determine whether to rebroadcast have difficulty in mobile environments; outdated 2-hop neighbor knowledge corrupts the determination of next-hop rebroadcasting nodes.

Given the inability of the Probability Based and Area Based methods to minimize the number of rebroadcasting nodes, protocols in these two categories fail to operate efficiently in congested networks. We could improve the protocols in these two categories by adapting the protocol to a node's neighbor count and local congestion level. The implementation of this improvement, however, would require the addition of both "Hello" packets and an adaptive RAD. Unfortunately, these extensions negate the inherent simplicity of the protocols, which is their most attractive feature. Furthermore, if "Hello" packets are used, a protocol which makes more intelligent use of the obtained neighbor knowledge is preferred.

We show that adapting RAD to the current congestion level improves SBA's performance in congested networks; at the highest congestion levels, the delivery ratios of Adaptive SBA were higher than AHBP. However, the gain in performance of our Adaptive SBA protocol requires a higher number of rebroadcasting nodes than AHBP. In addition, because SBA naturally adapts to dynamic topologies by increasing the number of rebroadcasting nodes, the RAD adaptive version of the protocol is unable to cope with a severe network environment (i.e., high node mobility, high node count and high bandwidth congestion).

AHBP-EX provides the best performance in the most severe network environment studied. Unfortunately, its high sensitivity to node mobility produces the lowest delivery ratio in networks where the environment is dominated by topological changes (see Section 4.3.3).

Generally, one would expect the trade off between algorithm complexity and redundant retransmissions to be a net gain; that is, power for more complex calculations is likely to be less than that required to transmit redundant packets. Combining the power ramifications with our performance results leads us to conclude that Neighbor Knowledge methods are preferred over other types of broadcast protocols. Unfortunately, there is no clear choice for a standard protocol between Adaptive SBA and AHBP-EX, as each fits a niche area. Therefore, we recommend that both protocols be considered for further improvements and standardization. In their current forms, AHBP-EX is recommended for semi-static topologies or extremely congested networks. Adaptive SBA is recommended for all other expected scenarios.

Finally, our performance evaluation of broadcast protocols provides direction for valuable future research. In Section 4.3.1 we demonstrate that AHBP approximates MCDS within 10% in dense networks. While CDS Based Broadcasting and other algorithmic optimizations provided in [44] and [45] may improve the approximation of MCDS, the fact that these protocols all rely on extremely accurate neighbor knowledge information makes them unattractive in dynamic networks; in other words, they share AHBP's deficiency in non-static networks. Therefore, future research in algorithmic optimizations should take a back seat to research in making these protocols effective in mobile networks.

Another important area for future research relates to SBA's RAD. Recall that SBA weights the RAD to a node's relative neighbor degree. We recommend that SBA's RAD determination be modified such that nodes with isolated neighbors rebroadcast first, similar to both AHBP and Multipoint Relaying. The goal of this proposed SBA modification is to minimize the number of rebroadcasting nodes and make it more effective in networks characterized by high congestion. Similarly, more research into adapting RAD to congestion might provide a more robust solution than the simple scheme we propose in Section 4.3.2.

REFERENCES

[1] C. Perkins and P. Bhagwat. Destination sequenced distance vector routing for mobile computers. *Computer Communication Review: Technical Newsletter of ACM Special Interest Group on Data Communication (SIGCOMM)*, 24(4):234–244, October 1994.

[2] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imelinsky and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[3] S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *ACM Journal on Wireless Networks*, 1(1):61–81, 1995.

[4] S. Murthy and J.J. Garcia-Luna-Aceves. A routing protocol for packet radio networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 86–95, 1995.

[5] V. Park and S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1405–1413, April 1997.

[6] Z. Haas. A new routing protocol for reconfigurable wireless networks. In *Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC)*, Oct. 1997.

[7] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM*

Conference on Communications Architectures, Protocols and Applications (SIGCOMM), pages 234–244, 1994.

[8] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[9] G. Pei, M. Gerla, and X. Hong. LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 11–18, 2000.

[10] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communications*, 4(2):1–36, 1997.

[11] R. Dube, C.D. Rais, K.-Y. Wang, and S.K. Tripathi. Signal stability based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communications*, pages 36–45, February 1997.

[12] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 243–254, 2000.

[13] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communications*, pages 48–57, February 2001.

[14] Y. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 66–75, 1998.

[15] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 76–84, 1998.

[16] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu. Grid: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18(1):37–60, 2001.

[17] C. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90–100, 1999.

[18] K. Ji and S. Corson. A lightweight adaptive multicast algorithm. In *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, pages 1036–1042, 1998.

[19] M.S. Corson and S.G. Batsell. A reservation-based multicast (RBM) routing protocol for mobile networks: Initial route construction phase. *ACM/Baltzer Wireless Networks*, 1(4):427–450, 1995.

[20] C. Chiang, M. Gerla, and L. Zhang. Shared tree wireless network multicast. In *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, pages 28–33, 1997.

[21] C. Chiang, M. Gerla, and L. Zhang. Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. *ACM/Baltzer Cluster Computing*, 1(2), 1998.

[22] C. Chiang and M. Gerla. On-demand multicast in mobile wireless networks. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 1998.

[23] C. Chiang, M. Gerla, and L. Zhang. Adaptive shared tree multicast in mobile wireless networks. In *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, pages 1817–1822, 1998.

[24] M. Gerla, C-C. Chiang, and L. Zhang. Tree multicast strategies in mobile, multihop wireless networks. *ACM/Baltzer Journal of Mobile Networks and Applications (MONET)*, 4(2), 1999.

[25] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand multicast routing protocol. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, September 1999.

[26] J.J. Garcia-Luna-Aceves and E.L. Madrga. A multicast routing protocol for ad-hoc networks. In *Proceedings of the IEEE Computer and Communications Societies (INFOCOM)*, pages 784–792, 1999.

[27] E.L. Madruga and J.J. Garcia-Luna-Aceves. Multicasting along meshes in ad-hoc networks. In *Proceedings of the IEEE International Conference on Computer Communications (ICCCN)*, June 1999.

[28] C. Perkins and E.M. Royer. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 207–218, 1999.

[29] C. Perkins, E. Royer, and S. Das. Ad hoc on demand distance vector (AODV) routing. Internet Draft: draft-ietf-manet-aodv-09.txt, November 2001.

[30] Z. Haas, M. Pearlman, and P. Samar. The interzone routing protocol (IERP) for ad hoc networks. Internet Draft: draft-ietf-manet-zone-ierp-01.txt, June 2001.

[31] IEEE Standards Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In *IEEE 802.11 Standard*. IEEE, New York, 1997. ISBN 1-55937-935-9.

[32] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. In *Proceedings of the International Workshop on Discrete Algorithms and methods for Mobile Computing and Communication (DIAL M)*, 1999.

[33] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson. A simple protocol for multicast and broadcast in mobile ad hoc networks. Internet Draft: draft-ietf-manet-simple-mbcast-01.txt, July 2001.

[34] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 1999.

[35] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.

[36] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2000.

[37] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 1975.

[38] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA - Rapport de recherche, 2000.

[39] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol (OLSR). Internet Draft: draft-ietf-manet-olsr-05.txt, October 2001.

[40] W. Peng and X. Lu. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Science and Technology - Beijing, China*, 2002.

[41] W. Peng and X. Lu. Efficient broadcast in mobile ad hoc networks using connected dominating sets. *Journal of Software - Beijing, China*, 1999.

[42] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. CAIP Technical Report 248 - Rutgers University. http://www.caip.rutgers.edu/˜marsic/mobile/ (Accessed on July 1, 2001).

[43] M. Srivastava. Tutorial 5 energy efficiency in mobile computing and networking. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 2000.

[44] J. Wu and H. Li. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. In *Proceedings of the International Workshop on Discrete Algorithms and methods for Mobile Computing and Communication (DIAL M)*, 1999.

[45] M. Seddigh, J. Gonzalex, and I. Stojmenovic. Internal node based broadcasting in wireless networks. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, 2001.

[46] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. In *Proceedings of the European Symposium on Algorithms (ESA)*, 1996.

[47] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. Multi-hop wireless ad hoc network routing protocols. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 85–97, 1998.

[48] J. Boleng. Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks. In *Proceedings of the IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, 2001.