Distance ADaptive (DAD) Broadcasting for Ad Hoc Networks

Xiaohu Chen, Michalis Faloutsos, Srikanth Krishnamurthy
Department of Computer Science
University of California, Riverside
{*xchen, michalis, krish*}*@cs.ucr.edu*

***Abstract*—In mobile ad hoc networks, it is often necessary to broadcast control information to all the constituent nodes in the network. Possible applications include searching for a destination node (as a part of routing) or a particular service such as DNS look-up. Flooding, which is often deployed to achieve the above objective, is expensive in terms of overhead and wastes valuable resources such as bandwidth and power. An improvement to flooding is to choose probabilistically a subset of nodes to rebroadcast. In this paper, we propose to use the signal-strength to improve the efficiency of broadcasting. We propose a protocol to select a set of nodes for rebroadcasting on the basis of their relative distance from the previous broadcast. We show how we can use the signal strength as an estimate of the relative node distance. Our simulations indicate that our approach can increase the efficiency of broadcasting significantly. Our simulations show that we can achieve the same coverage with approximately 20% less rebroadcasts. In addition, the time taken by the global broadcast is also reduced by more than 20%.**

I. INTRODUCTION

Several emerging applications such as sensor networks and battlefield communications will rely on ad hoc networks. Ad hoc networks consist of static or mobile nodes which act both as users and routers. The nodes communicate with wireless links over a shared medium. The limited power that the mobile nodes have introduces power-conservation as a fundamental requirement for these networks.

In this paper, we examine methods to improve the efficiency of broadcasting in an ad hoc network. Broadcasting will be a significant building block for the efficient function of these networks, especially given the volatile nature of the network. For example, nodes will need to search for a service such as a DNS look up or for a routing path. Broadcasting is power intense and can threaten the life duration of the network. We define a **broadcast session** to be the process of one node exploring a part of the network. Within a broadcast session, many nodes may **rebroadcast** packets to support the particular broadcast session. Our goal is to perform this broadcasting in the most efficient way. The primary metric that determines the broadcast efficiency is the power consumed. The number of packet transmissions dominates the power consumption, since a transmission requires orders of magnitude more power than most other common functions such as local processing.

The problem of broadcasting has received relatively little attention. A straightforward but not efficient method is to *flood* the network. In flooding, every node rebroadcasts every broadcast packet it receives. Flooding guarantees that all connected nodes will "hear" a particular broadcast, if we assume no collisions. For a dense network, flooding can be very inefficient and can cause significant contention and collisions also known as the broadcast storm problem. Recently, Ni et al.[1] proposed schemes to improve the efficiency of a broadcast. They propose to select randomly a percentage of the neighborhood nodes to perform rebroadcasting.

We propose to use a power-aware scheme to improve the broadcasting efficiency. Simply put, our scheme attempts to select intelligently the nodes that will rebroadcast: we "encourage" nodes that are far from the previous rebroadcast node. Our scheme utilizes the signal strength and neighborhood density to determine the set of nodes that perform the rebroadcasting. Simulation results show that our approach can improve the broadcasting efficiency significantly. More specifically, our protocol can reduce the number of rebroadcasts by 20%. Furthermore, the latency incurred in the broadcasting procedure can be reduced: the completion is faster by 20% while it covers 13% more nodes.

The remainder of this paper is organized as follows. Section 2 presents the background and the models we use in this paper. The metrics of interest are also listed in Section 2. Section 3 describes the scheme of outmost broadcasting and also gives out the algorithm. Simulation result and discussion are showed in section 4. Section 5 concludes our work.

II. BACKGROUND AND MODEL

We define the goal of a broadcast session to be "to reach a part" of the network, but not necessarily the entire network. We prefer this more general definition, since this could reflect more accurately, the intentions of an application. In some cases, a search among a certain percentage of total nodes may be satisfactory for a particular application. For example, when a node wants to contact a server, it only needs to find the nearest one within several hops. With this definition, we let the application or the user select the appropriate extent of the broadcast. The requirement is to reach the desired scope with as few rebroadcasts as possible.

We make some typical assumptions about the underlying network. The nodes share a single common channel with carrier sense multiple access (CSMA with no collision detection). Furthermore, we assume that each node has the same transmission power. We assume that each node maintains information about its neighborhood in a table. We can have two different mechanisms to update the table. First, the neighborhood table

This material is based upon work supported by the DARPA award FTN F30602-01-2-0535

is updated through a periodic HELLO message[1]. Second, it can be updated whenever a node hears a packet transmission.

To improve on flooding, a sub set of nodes can be selected to perform rebroadcast. Ni et al. [1] presented several different ways to reduce the number of rebroadcasts. One of their proposed methods was a probabilistic scheme, where they encourage only a certain number of nodes in the neighborhood to rebroadcast. We compare our protocol with this one and for this we present it in detail below.

In addition, several other mechanisms were proposed to reduce rebroadcasts such as counter based, distance based and also GPS for position location[1]. Other mechanisms can also be used to control the message overhead such as TTL (time-to-live) and expanding ring search.

We briefly describe the general probabilistic broadcast (**GEN**) algorithm proposed in [1]. A parameter **k**, which we refer to as the target rebroadcast size, is specified. It represents the average number of neighboring nodes that are required to rebroadcast. Each broadcast packet carries the size of the neighborhood of the sender. When a node receives the packet, it rebroadcasts the packet according to the following procedure. It randomly generates a number **n** between 0 and the neighborhood size of the sender of the packet. If the number $n < k$, it will rebroadcast. It is easy to see that the protocol attempts to have 'k' new rebroadcasts after each rebroadcast. A node that has already rebroadcasted once for a particular broadcast session will not broadcast again.

We need to define performance metrics in order to compare the efficiency of the various broadcast schemes. We use the following metrics:

- **Coverage**: is the fraction of nodes of the network that are reached in one broadcast session (This includes many rebroadcasts from nodes other than the initiator).
- **Broadcast Efficiency** : is the average number of newly reached nodes per rebroadcast packet. The broadcast efficiency is equal to total number of reached nodes divided by the total number of rebroadcast[2] packets. Note that for the first broadcast the efficiency is equal to the size of the neighborhood.
- **Broadcast Latency or Duration**: is the duration of the broadcast session, i.e., the time interval between the first and last rebroadcast.

Note that the above metrics are related. For high coverage, we naturally need more rebroadcast packets. Hence, we introduce the efficiency metric; to count the number of nodes we reach (gain) versus the rebroadcasts (cost).

III. **Distance-Aware Broadcasting**

The key idea of this work is to estimate and use the distances between nodes in order to improve the efficiency of a single broadcast session. Intuitively, we want to maximize the coverage while reducing wasted rebroadcasts. More specifically, we propose to use both signal information and neighborhood size to choose the set of nodes that rebroadcast. We want the outmost neighboring nodes to rebroadcast. Figure 1 illustrates the concept. Assume that node 's' is the node that initiates the broadcast session. If the outmost nodes 4, 5, 6, 7, 8 rebroadcast, we maximize the number of new nodes we can reach. Broadcasts of nodes 1, 2 and 3 do not yield much if the aforementioned nodes rebroadcast. Thus, we do not want nodes 1, 2, 3 to rebroadcast.

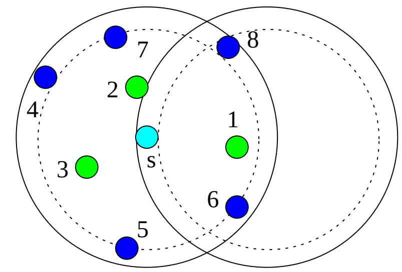

Figure 1. Broadcasting and outmost nodes

The natural question is: How can we identify the outmost nodes without introducing a significant increase in overhead? Clearly, if nodes are equipped with GPS, then, we can have accurate measurements of their relative distances. However, GPS may be too expensive or power-consuming. We propose the use the signal strength to estimate the relative distance between nodes[2].

When a node receives a packet, it can measure the received signal strength of that packet. By knowing the signal strength and channel model of the link to the receivers, node can estimate the distance to the sender of the packet[3]. A neighbor table is maintained as we described in Section II. The difference, however, is that now we also keep the signal strength from the most recent packet received from each neighbor. Each entry in the table has two fields: the node id and the received signal strength. Entries are sorted in ascending order of signal strength. Thus, the furthest node correspond to the first entry, and the n-th furthest node corresponds to the n-th entry.

Before we describe our approach, let us examine a simpler and more naive way of performing the broadcast. Let us assume that we allow nodes only beyond a certain distance to rebroadcast for all the rebroadcasts of a session. This would not be as effective, since it could lead to an early termination of the broadcast. There may be cases where there are no neighbors beyond the pre specified distance. In that case, there will be no further rebroadcasts if this naive scheme is used in that particular neighborhood.

We propose to use the distance metric in conjunction with a "required" number of rebroadcasts. Consequently we can think of two Distance-ADaptive schemes. In the first scheme, we specify a certain number of outmost nodes (**DAD-NUM**), while in the second, we specify a percentage of the outmost nodes (**DAD-PER**) that we select to perform the rebroadcast.

[1] For some ad hoc routing protocols, the periodic message is actually needed[3,4].

[2] The term rebroadcast here includes also the first broadcast packet sent by the initiator of the broadcast.

[3] For Example, we can use $P_r = cP_t(1/d^n)$ to calculate the distance d with known P_r and P_t

Protocol DAD-NUM: The input parameter is the number k of outmost nodes that we want to perform rebroadcast.

- Before a node rebroadcasts, it consults its neighborhood tables, and finds the threshold signal-strength value S_{thres} such that there are k values less than this value S_{thres}. In other words, there are k neighbors further than the distance that corresponds to S_{thres}. If a node has neighbor size of less than k, we set the S_{thres} as maximum of the signal strength values, i.e., all of the nodes neighbors will be selected to perform rebroadcast. Each rebroadcasting node includes the threshold signal-strength value in the broadcast packet.
- Once a node receives a broadcast packet, it records the received signal strength S_{rec}. Furthermore, it retrieves the value S_{thres} from the packet. It compares the two signal-strength values. If $S_{rec} > S_{thres}$, it ignores the packet. Otherwise it rebroadcasts.

Protocol DAD-PER, it is similar to the previous protocol, with the only difference being that we select a percentage p of nodes that we encourage to rebroadcast, instead of an absolute number. Thus, the threshold is specified by the value of the last of the p% of the top entries in the neighborhood table. It turns out that this method performs as well as the distance-unaware approach GEN. However, it does not perform as well as the previous, scheme DAD-NUM.

IV. SIMULATION RESULTS

We use GLOMOSIM[5] to perform our simulations. We have modified the implementation of the 802.1l MAC protocols to simulate the CSMA/CA behavior. HELLO messages are implemented in the protocol to allow the exchange of neighborhood information, i.e., each node will broadcast a HELLO message to notify its neighbors of its presence periodically. So each node has a knowledge of its neighbors. We assume that the transmission range is symmetric. The periodicity of the HELLO messages is 5 seconds and the rebroadcast jitter time (packets wait a short period of time before sent to MAC layer) is set to 10 msec.

In the simulation models, the network is deployed in a 3000m x 3000m area. The transmission radius of a node is fixed to 223m. The nodes are randomly distributed. We record the number of nodes reached in a broadcast session, the corresponding number of rebroadcast packets sent, as well as the latency incurred. For each configuration, we perform 200 iterations and compute the average result. Simulation is also performed with various mobile speeds and with various nodes densities.

A. Broadcast efficiency of different broadcasting schemes

We first compare the broadcast efficiency of the two schemes. Since DAD-PER uses a percentage to decide the number of nodes that will rebroadcast, its efficiency is a function of percentage. In order to compare with the efficiency of DAD-NUM, the efficiency of DAD-PER is normalized as a function of corresponding number 'k'. The conversion uses the product of the average neighbor size and percentage to get the number of rebroadcasting nodes 'k'. The simulation is performed on a topology with 500 nodes, and the result in terms of performance are shown in figure 2.

The figure shows that DAD-NUM performs best for all 'k's. When k is small, DAD-PER is slightly better than GEN but it is not as good as GEN when k is greater than 3. The reason is that for those nodes that have a large neighbor size, DAD-PER will cause a large number of nodes to rebroadcast. However, for GEN and DAD-NUM, statistically, only a certain number of nodes will rebroadcast no matter how big the neighbor size. Pure flooding has the least efficiency of 1 for all values of 'k's.

Note that the broadcast efficiency is only one metric that we use to evaluate a broadcasting scheme. We also consider coverage. Figure 2 shows that when k is 1, the broadcast efficiency is as high as 7. However, the corresponding coverage is only 1.6%. The efficiency is only useful when the coverage is large enough.

From our result, we observe that the coverage of DAD-NUM and DAD-PER are very close and DAD-NUM has a higher efficiency than DAD-PER. So in the rest of the paper, we only use DAD-NUM and compare with the general scheme (GEN).

B. Coverage and latency

We next compare the behaviors of DAD-NUM and GEN. The efficiency of two schemes are compared in figure 2. When rebroadcast size increases, we see that DAD-NUM is always better than GEN.

The coverage of DAD-NUM and GEN is compared in figure 3. The simulation results show that DAD-NUM can cover more nodes than GEN. As the number 'k' increases, the coverage increases and is close to the coverage provided by pure flooding (represented by lines). The increase on coverage provided by DAD-NUM over GEN is represented by the bars in the graph. The gain can be up to an increased coverage of 20% of the total nodes.

The advantage of DAD-NUM is that it can achieve large coverage than GEN while attaining a higher broadcast efficiency. Figure 4 shows the relation between the efficiency and coverage for DAD-NUM and GEN. For achieving a set coverage, DAD-NUM is more efficient than GEN, thus, requires less rebroadcast packets.

Another advantage of DAD-NUM is that it takes a lower time to complete the broadcast session than GEN. Since the outmost nodes perform rebroadcast, each rebroadcast will cover a large area. Thus, a smaller number of rebroadcast relays are needed to finish the whole broadcast session. Figure 5 compared the delays of DAD-NUM and GEN. When k is small, the latencies are similar because of the similar low coverage. As k increases, the advantage of DAD-NUM becomes obvious. When k=8, the latency of the broadcasting procedure of DAD-NUM is lower by 21% while an increase of 13%. Since, too small coverage is not interesting for real applications, we use a rebroadcast size k=7 in the rest of our discussion.

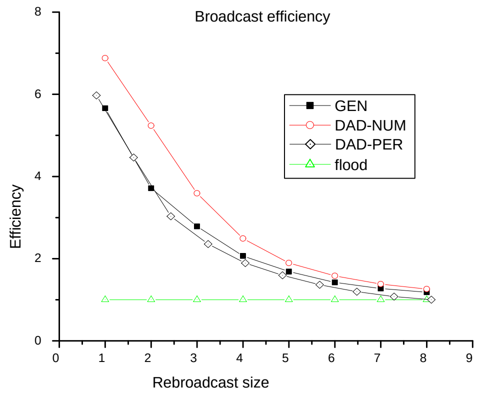

Figure 2. Broadcast efficiency for different schemes (Efficiency of broadcast vs. rebroadcast size)

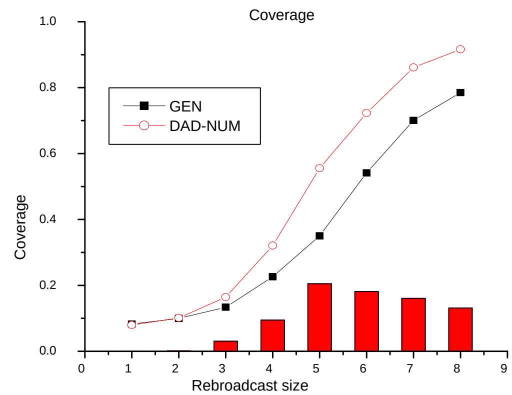

Figure 3 Coverage of DAD-NUM and GEN vs. rebroadcast size (Additional coverage of DAD-NUM over GEN is represented as bars. When k=8, additional coverage is 13%)

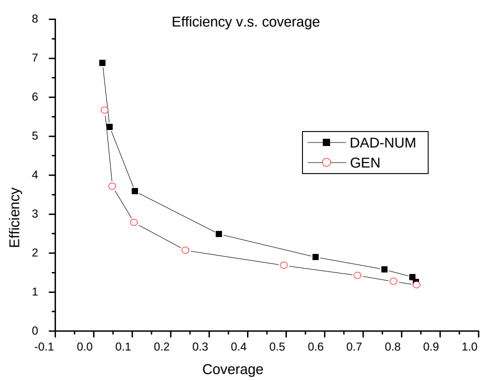

Figure 4 Efficiency of DAD-NUM is high even with high coverage. Observe that DAD-NUM covers nodes with better efficiency.

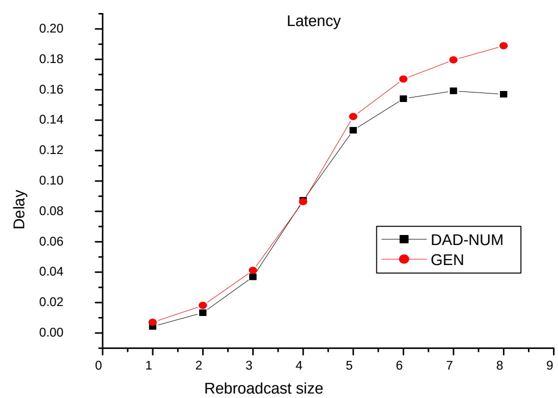

Figure 5 Compare of latency for different size k (When k=8, GEN's latency is 21% longer)

C. *Effect of mobility*

In a mobile network, nodes are moving around with different velocities. The motion of the nodes will impact a broadcast session in various ways. However, the behavior of broadcasting actually does not change much with different mobility patterns.. The reason is that the duration of a broadcasting session is not very long. As Figure 5 shows, the delay of a broadcast session is about 150 millisecond. With a speed of 10m/s or 20m/s, the actually position of nodes do not change by much during the session. Since the behavior is very similar for other speeds, we simply use a speed of 20m/s for further experiments.

D. *Effect of nodes density*

We have considered a network with 500 nodes and analyzed the results. A change in node density can lead to some other interesting observations. As the node density increases, the average neighbor size increases. This will result in high contention among neighboring nodes. Our scheme, thus, is impacted by two aspects. First, our scheme only uses a small fraction of the nodes to perform the broadcast. If the density of the nodes increases, each rebroadcast packet can potentially reach a large number of nodes. So the efficiency increases. Second, the coverage is slightly degraded due to the contention.

We simulated four different topologies of varying size. They contain 500, 600, 800, 1000 nodes, respectively. The corresponding average degree of a node in the specified cases are 8.0, 10.4, 12.9 and 16.1, respectively. As figure 6 shows, the efficiency of broadcasting increases steadily as the node density increases (represented by bars). The coverage is slightly degraded (represented by lines). For the 1000 nodes case, the efficiency is around 3, which corresponds to a coverage of about 80% with a cost of only 25% of that incurred in pure flooding. We use a large value of k (=7) since this is the regime of

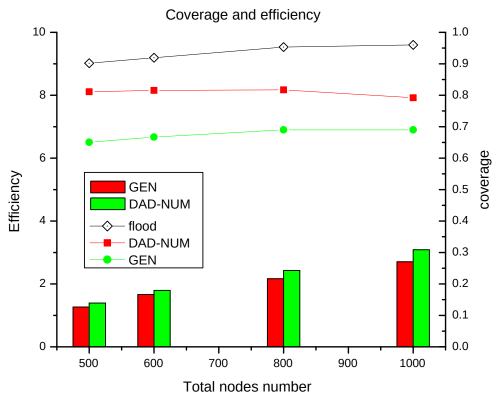

Figure 6 Coverage of DAD-NUM and GEN and flood is showed with lines. Efficiency of DAD-NUM and GEN is represented by bars.

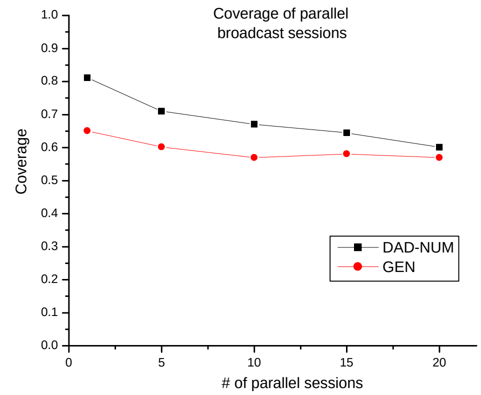

Figure 7 Coverage for different number of parallel broadcast sessions.

interest.

E. *Effect of multi-sessions*

So far all the simulations that we have shown are for the case wherein the network is lightly loaded and there is only one ongoing broadcast session at a given time. In real applications, multiple broadcasting sessions could arise in different parts of the network.

We compared the coverage of DAD-NUM and GEN when multiple broadcast sessions exist simultaneously. These sessions are randomly generated. Figure 7 shows the results of our simulations.

As the number of broadcast sessions increases, the coverage decreases because of an increase in contention level. When the number of parallel broadcast sessions are large, both GEN and DAD-NUM behave similarly which means that they are impacted in the same manner.

V. **Conclusion**

The use of signal-strength is a novel idea that is worth examining further to establish its full potential. Our initial simulations suggest that there may be significant advantages in performing signal strength adaptive broadcast.

As our main contribution, we propose to use the signal strength to improve the efficiency of broadcasting and develop protocols to do so. We show that we can use the signal strength of receiving packet to infer the relative distances among nodes. Using this we ensure that only the outmost neighbors of a broadcasting node perform the rebroadcast. In an ad hoc network, power consumption is critical, and for this, we require the minimization of the number of packet transmissions.

Our work can be summarized as follows:

- We show that broadcasting can be improved greatly by choosing the outmost nodes within a broadcasting nodes range for rebroadcasting. Our protocol requires 20% less rebroadcasts for covering the same area compared to previous methods.
- We demonstrate that signal-strength can be used effectively to improve the efficiency of broadcasting.
- As an added bonus, by using our approach the duration of a broadcast session procedure can be reduced. In our simulations, the broadcast is faster by about 20% while it covers 13% more nodes.

Future work. We want to integrate our approach in an adaptive framework wherein the user will define the extent of the broadcast. The mechanism will then self-configure to achieve the desired broadcast. The framework would adaptively fine-tune the broadcast parameters according to the real application. We would like to stress-test our approach in highly mobile environments. We also want to find its sensitivity to potentially misleading or obsolete neighborhood information. For both these cases, we would like to develop robust schemes to deal with mobility with information staleness.

References

[1] S.Y. Ni, Y.C. Tseng, Y.S. Chen and J.P. Sheu, "The broadcast storm problem in a mobile ad hoc network," Proc. ACM/IEEE MobiCom, August 1999.

[2] K. Pahlavan, P. Krishnamurthy, Principles of Wireless Networks, Prentice Hall, 2002.

[3] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", ACM Mobile Networks and Applications Journal, Special issue on Routing in Mobile Communication Networks, 1996.

[4] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In ACM SIGCOMM Symposium on Communication, Architectures and Applications, 1994.

[5] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory. GloMoSim: A scalable simulation environment for wireless and wired network systems.http://pcl.cs.ucla.edu/projects/glomosim, 2000.

[6] R. Dube, C.D. Rais, K. Wang, and S.K. Tripathi, "Signal stability based adaptive routing (SSA) for ad hoc mobile networks, " IEEE Personal Communication, Feb. 1997.