An Efficient Distributed Network-Wide Broadcast Algorithm for Mobile Ad Hoc Networks

John Sucec and Ivan Marsic
CAIP Center, Rutgers University
Piscataway, NJ 08854

***Abstract*--In this paper, an algorithm for efficient network-wide broadcast (NWB) in mobile ad hoc networkds (MANETs) is proposed. The algorithm is performed in an asynchronous and distributed manner by each network node. The algorithm requires only limited topology knowledge, and therefore, is suitable for reactive MANET routing protocols. Simulations show that the proposed algorithm is on average 3-4 times as efficient as brute force flooding. Further, simulations show that the proposed algorithm compares favorably over a wide range of network sizes, with a greedy algorithm using global topology knowledge, in terms of minimizing packet transmissions. The application of the algorithm to route discovery in on-demand routing protocols is discussed in detail. Proofs of the algorithm's reliability and of the intracatability of solving for a minimum sized transmitter set to perform NWB are also given.**

I. Introduction

Network-wide broadcast (NWB) is an essential feature of some of the emerging on-demand routing protocols in mobile ad hoc networks (MANETs). MANETs are multiple-hop, packet-forwarding networks consisting of potentially mobile nodes interconnected by wireless links. Since for on-demand routing protocols nodes do not proactively maintain forwarding tables for destinations lying beyond some small radius that is less than the network diameter, nodes must routinely perform a reactive route discovery procedure. As implemented in the Ad hoc On-demand Distance Vector (AODV) [1] and Dynamic Source Routing (DSR) [2] protocols, route discovery involves disseminating a route request (RREQ) packet throughout the network. Without an efficient NWB algorithm, such a route discovery procedure results in considerable packet overhead.

Both AODV [1] and DSR [2] have timeout mechanisms that allow the protocol to reinitiate a route query if the previous attempt fails. Thus, the route discovery process for these protocols assumes a potentially unreliable NWB procedure. A given NWB may fail for a number of reasons including partitioning of the network and packet collisions due to the shared media nature of the network interfaces. (Since network partitioning can cause NWB to fail for any protocol, this paper will consider only networks that are connected.) In any event, if a particular RREQ packet fails to reach the target node (T), or a node that has a path to T in its route cache, then the route discovery attempt will be declared a failure and reinitialized by the source node (S).

Due to the retry measure of the on-demand route discovery process, it is evident that although an unconditionally reliable NWB implementation is desirable, it is not essential. This is a useful property of the AODV and DSR protocols. Instead of implementing a NWB procedure that is proven to be 100% reliable in that it guarantees packet delivery to all network nodes within some bounded time, one need only to implement a procedure that in the absence of collisions or network partitions can be shown to be reliable. Thus, with the requirement of unconditionally reliable NWB relaxed for on-demand route discovery, a NWB procedure that requires modest topology knowledge and yet achieves high operational efficiency can be considered.

Minimizing the topology information required by the NWB procedure is crucial for AODV and DSR because neither protocol proactively acquires topology information beyond a single hop radius of a given node (DSR actually specifies zero proactive acquisition of topology data). This means the NWB procedure must be able to function with considerably less than complete topology knowledge. Further, due to the fact that AODV and DSR are routing protocols for flat (non-hierarchical) network topologies, topology structure built into networks with hierarchical or clustering properties, as in [8,10], cannot be exploited for NWB. Lastly, both protocols have minimal periodic routing message overhead and it is desirable not to substantially increase the existing volume of periodic messaging.

In terms of performance, a new NWB procedure should minimize the number of transmissions needed to disseminate a given packet (e.g., a RREQ packet) to every network node. The number of transmissions should obviously be something less than $|V|$ which corresponds to the number of transmissions required by brute force flooding[1]. In contrast, an ideal NWB implementation would maximize the number of non-transmitting nodes. Thus, the metric used here to assess the *efficiency gain* (ε) of a NWB algorithm is defined as follows:

$\varepsilon \equiv$ Number of transmitters required by brute force flooding (i.e., $|V|$) divided by the average number of transmissions required by the NWB algorithm.

As simulation results will show, the NWB algorithm proposed in this paper provides substantial gains in efficiency over brute force flooding.

The remainder of this paper is organized as follows. Section II describes previous work relevant to NWB in packet radio networks. In Section III, a new *asynchronous* and *distributed* NWB algorithm, called lightweight and efficient network-wide broadcast (LENWB), is proposed that is particularly well suited for on-demand routing protocols in MANETs. Section IV reports on Monte Carlo simulations of LENWB that were performed to assess the efficiency of the new algorithm. Section V discusses the simulation results. Lastly, Section VI provides some conclusion and presents ideas for continuing work.

II. PREVIOUS WORK

A considerable number of references are relevant to the problem of NWB in radio networks. Among these include [3-7,9-11,14,15,18].

References [3,5,14,15,18] discuss tree-based algorithms. In [3,5,18], efficient distributed algorithms are proposed for constructing group shared or source based spanning trees, which can then be used to support messaging such as NWB. Reference [5] additionally specifies how time slots can be assigned to non-leaf nodes of the constructed tree for collision-free transmissions. In [14], integration of a multicast protocol with a MANET routing protocol is proposed. The purpose being that coupling between the multicast and unicast routing procedures will improve efficiecny for multicast operations (including NWB). Reference [15] presupposes a spanning tree rooted at each NWB and goes and describes a scheduling procedure for reliable in-order delivery of NWB packets. A drawback to all tree-based approaches is the messaging overhead associated with construction and maintenance of the spanning trees. Tree-based techniques for NWB, therefore, will not be considered further in this paper.

In [4], an efficient distributed algorithm for assigning TDMA time slots to network nodes, requiring only two-hop topology knowledge, is proposed to ensure collision-free transmissions. Such a contention resolution scheme ensures that the NWB is reliable. Nevertheless, this approach by itself does not attempt to reduce the number of nodes required to disseminate the NWB message.

In [6], the issue of achieving reliable NWB while minimizing the delay to complete the NWB is considered. However, the algorithm described therein, requires each node to have global network topology knowledge. Such topology information will not typically be available to nodes in MANETs where an on-demand routing protocol is employed.

In [11], the Zone Routing Protocol (ZRP) is described. Like AODV and DSR, the ZRP is a MANET routing protocol. Unlike AODV and DSR, however, the ZRP is a *hybrid on-demand* routing protocol. Hybrid in that each node running the ZRP protocol proactively maintains topology information about nodes up to *h* hops distant. When a route to a node beyond this *h*-hop routing zone is required, the node initiates a route discovery process similar to that of AODV and DSR. To make the route discovery process more efficient, ZRP employs a technique called *bordercasting* which a source node (*S*) uses to efficiently disseminate a RREQ packet to its peripheral nodes by multicasting within its routing zone. Although, the bordercasting approach to NWB efficiently disseminates the RREQ packet within a given node's routing zone, it results potentially in many redundant packet transmissions by the border node recipients. As will be seen in Section III, the LENWB algorithm is more efficient than bordercasting for NWB because all nodes (not just interior nodes of a routing zone) will block propagation of a RREQ packet if there is a more efficient set of local nodes to disseminate it.

In [9], another reliable NWB solution is described. The approach here relies on a clustering structure within the network to efficiently disseminate a given packet cluster members via the clusterhead nodes. Acknowledgement messages (ACKs) received from cluster members assure the clusterhead nodes that the packet has been successfully received by all of its

[1] Where, V is the set of nodes in a given network, E is the set of bi-directional links connecting nodes in V and $G = (V,E)$ is the undirected graph formed by V and E. $|S| \equiv$ Cardinality of the set S.

cluster members. The acknowledgement information is then forwarded to the clusterhead predecessor lying in the path leading back to the source node of the NWB. Although not discussed in [9], it can be modified so as to be more efficient by removing the reliability requirement. Thus, ACK messages need not be returned. The resulting implementation is then similar to the dissemination of NWB packets performed by the Cluster Based Routing Protocol (CBRP) [10] and possible with the adaptive clustering protocol presented in [13], where only clusterhead nodes and gateway nodes need to forward the NWB packet. However, a mechanism in the routing protocol is required for cluster creation/maintenance and clusterhead election.

The CBRP specified in [10] merits further discussion. First, the document specifically discusses in detail how the clustering of nodes can be used for efficient NWB, in particular for the application of route discovery. Second, unlike [9,13] and earlier papers on clustering protocols [16,17] where only single hop topology knowledge is presupposed, CBRP presumes two hop topology knowledge as does the LENWB algorithm proposed herein. However, the CBRP and the LENWB algorithm fulfill different roles in the protocol stack. LENWB is designed to function as a single procedure to be called upon by an overlying routing protocol such as AODV, DSR or even ZRP. Effectively, the LENWB algorithm is to serve as an enhancement or extension to some of the existing routing protocols. The CBRP, on the other hand, subsumes a source routing protocol for basic routing operations and utilizes the cluster formation portion of the protocol to facilitate efficient NWB. Thus, although the CBRP and the LENWB algorithm share a similar objective and require similar topology knowledge, the architectures of these two approaches are quite different. Further, while both proposals require two-hop topology knowledge, CBRP requires additional communication and maintenance overhead due to data structures related to electing cluster heads, determining gateway nodes and identifying neighboring clusters, which are not needed in a LENWB implementation. Therefore, unless the clustering of CBRP is to be also used for the channel assignment, the control overhead of CBRP is not justified for the purpose of NWB.

In [7], several heuristics are proposed to mitigate the effect of a broadcast storm. Unlike [3-6,9-11,14,15,18], minimal topology information is required as each network node decides independently, based on one or more heuristics, whether it should serve as a transmitter in the NWB. The basic idea of each of the proposed techniques is that upon receipt of an additional copy of a packet belonging to a NWB, the fewer will be the expected number of neighboring nodes that have not received a copy of the packet. If the expected area of coverage by previous transmissions within a given node's transmission radius exceeds some threshold, the node decides it no longer must propagate a copy of the packet and deletes it from its transmission queue. The heuristics evaluated in [7] include probabilistic, counter-based, distance-based and location-based. In particular, when the location-based approach is coupled with GPS coordinate information on neighboring nodes, this technique can be very powerful in minimizing the number of transmissions required for NWB and yet still providing high reliability. The drawback, of course, is that in the absence of GPS coordinate information, the proposed heuristics can not provide reliable NWB even in a collision free environment. Further, the probabilistic methods of [7] require tight handshaking between the MAC layer and upper layer protocols to delete a NWB packet from the transmission queue in the event that receipt of a redundant copy of the packet results in the expected area of coverage threshold being exceeded.

III. LENWB ALGORITHM

The LENWB algorithm presupposes that the routing protocol incorporates a form of Hello protocol where nodes periodically advertise their presence. The associated Hello packet transmitted by a given node, call it node *v*, is assumed to contain the following data items:

i. The node ID or IP address of node *v*.
ii. The degree of *v*, i.e., the number of neighbors of node *v*, *deg(v)*.
iii. The node ID or IP address of each neighbor of *v*.
iv. The degree of each neighbor of *v*.

Since data item (ii) can be easily computed from (iii), it can be considered as an optional field.

As discussed shortly in the LENWB algorithm description, (ii) and (iv) are used by nodes to assess their transmission priority relative to other nodes. Items (i) and (iii) are used by nodes to determine their two-hop topology. The two-hop topology about *v*, for the purposes of this paper, consists of all nodes and links belonging to the set of two-hop paths on which *v* lies. Nodes that can be reached from *v* by a one-hop path are the neighbors of *v*, or one-hop nodes of *v*.

Nodes that are in the two-hop topology of v but are not neighbors of v are two-hop nodes of v. Thus, although v is cognizant of all links between itself and its neighbors and of links between its neighbors and its two-hop nodes, v does not know of the links between pairs of its two-hop nodes. Links between a pair of two-hop nodes of v lie on paths with v that consist of three hops and, therefore, are not part of the two-hop topology information employed by the LENWB algorithm. Such additional topology information could be conveyed in the Hello messages with the overhead of not more than K^2 bits, where K is the maximum node degree in the network. However, the version of the LENWB procedure presented in this paper does not presuppose knowledge about links between pairs of two-nodes of v.

The following definitions are relevant to the algorithm's description and subsequent discussion in this paper:

- $v \equiv$ Node ID of an arbitrary network node at which LENWB is being run.
- $N(v) \equiv$ Set of neighbors of v.
- *nwb_id* $\equiv$ ID of a particular packet that requires NWB service.
- $u \equiv$ Forwarding node from whom a RREQ packet, with request ID *nwb_id*, has been received at v.
- $C(u) \equiv$ Set of nodes covered by the RREQ broadcast of u. Note, since v knows its neighbors' neighbors, $C(u)$ is known to v.
- $X \equiv$ Set of all transmitting nodes for a given RREQ instantiation.
- $C(X) \equiv$ Set of nodes covered by the transmissions of X. Successful NWB occurs when $C(X) = V$.
- $PC(w,v) \equiv$ Priority condition. A node w satisfies the priority condition with respect to the node v if one of the following is true:
  - $deg(w) > deg(v)$.
  - $deg(w) = deg(v)$ and $w < v$.
- $\pi \equiv$ A path denoted by the sequence of bi-directional links (edges), $\{(u,x_1),(x_1,x_2),\ldots,(x_{k-1},x_k)\}$ connecting node u with node x_k, such that each node x in the path lies not more than two hops from v, satisfies $PC(x,v)$, and each edge of the path has at least one terminal vertex in $N(v)$.
- $P \equiv$ Set of nodes such that for each $p \in P$, $PC(p,v)$ is true, p lies exactly two hops from v and $\exists$ a path π connecting u to p.
- $Q \equiv$ Set of nodes such that for each $q \in Q$, $PC(q,v)$ is true, $q \in N(v)$ and $\exists$ a path π connecting u to q.

For each unique instantiation of a NWB packet received by node v, the packet ID (*nwb_id*) and predecessor node (u) is maintained in a database record. Note, it is possible that v receives a copy of a given NWB packet from multiple, say k, neighbors. However, the structure of the database would remain essentially the same with a set of predecessor nodes, $\{u_1, u_2, \ldots, u_k\}$, being maintained. For the purpose of simplifying the discussion in this paper the LENWB algorithm implementation at each node will consider only the first forwarding node (u) from which a copy of a NWB instantiation is received. This provides a more rigorous condition under which to test the algorithm's performance and minimizes the amount of handshaking with the MAC layer protocol that might otherwise be required.

The LENWB algorithm, as described below, assumes reliable delivery of NWB packets to all neighbors of a transmitting node. In practice, however, successful receipt of NWB packets may be disrupted by packet collisions and/or node mobility that invalidates the neighbor status data available at a transmitting node. Implementation of the algorithm under the assumption that NWB packets may not be received by all neighbors may require extensions to enhance its reliability. Such extensions represent a direction for future work. The LENWB algorithm under the reliable delivery condition is as follows.

LENWB Algorithm:

1) If the received packet represents the first copy of a NWB packet with ID *nwb_id*, then write *nwb_id* and u into the appropriate fields of a NWB record and set $U = \{u\}$. Otherwise, disregard the redundant packet arrival and STOP.
2) If $N(v) \subseteq C(U)$, STOP.
3) Calculate P and Q.
4) Until $N(v) \subseteq C(U)$ or $U = \{u\} \cup P \cup Q$ do
   a) Select a node x satisfying $x \notin U$, $x \in (P \cup Q)$ and $x \in C(U)$.
   b) $U = U \cup \{x\}$.
5) If $C(U) \cap N(v) \neq N(v)$, queue the NWB packet for transmission. Else, discard the NWB packet.

Pseudo code for the combined processing specified in steps 3 through 5 of the algorithm description is given in Appendix A. Based on this pseudo code, it is evident that the local decision to block or transmit can be made in at worst $O(K^2)$ time where K is the maximum node degree of the network. A proof that the LENWB algorithm produces successful NWB under the condition of reliable packet delivery, is given in Appendix B.

Although the LENWB algorithm can be applied readily to any type of packet that requires NWB, the LENWB algorithm is of particular usefulness to on-demand routing protocols such as AODV and DSR. The LENWB algorithm would be married with the route discovery procedure of the on-demand routing protocol currently in operation. Specifically, the LENWB algorithm extends the decision-making process of whether a RREQ packet associated with a particular route query will be propagated by node v. For example, upon receipt of a RREQ packet, the route discovery module of the routing protocol may check whether an existing route to the target node T is in the route cache. If yes, a route reply is generated and the RREQ packet is not propagated. Otherwise, the RREQ packet is then processed by the LENWB algorithm to determine whether v needs to transmit the packet in order to cover $N(v)$. LENWB, therefore, may be treated as a form of *blocking process* that regulates propagation of RREQ packets at a particular node.

The algorithm description of course would have to be modified slightly to accommodate the route query *hop_limit* parameter that may be in effect for a particular AODV or DSR implementation. This could be easily achieved as follows. Step 1 would be augmented by requiring nodes to enter the *hop_limit* data into the record for packet *nwb_id*. Further, if *hop_limit* value is zero, then the algorithm stops. The Step 3 description would be amended to state: "Calculate P and Q such that all nodes in P and Q can be reached via a path π from u of not more than *hop_limit* hops." The additional restriction on P and Q ensures that v defers only to nodes not more than *hop_limit* hops from S (i.e., receive the packet with *hop_limit* > 0) and, therefore, can transmit the packet. Lastly, in step 5, the *hop_limit* field of packet *nwb_id* would be decremented, if it is queued for transmission.

In the event that collisions or node mobility result in not all nodes being covered by the NWB, it is possible that a copy of the RREQ packet will fail to reach T or a node cognizant of a valid route to T. For some applications, the failure for a NWB packet to reach all network nodes is unacceptable. For such applications, the LENWB algorithm is not suitable and one of the reliable NWB approaches discussed in Section II would be more appropriate. However, for the route discovery procedure employed in AODV and DSR, a NWB implementation that is not 100% reliable only means that the route discovery query will have to be initiated again. Thus, the LENWB algorithm is still very much suitable for the route discovery process.

IV. Simulation Results

Table I shows the four network topology conditions for which simulations were run. Simulation of the distributed operation of the LENWB procedure across all network nodes was based on a wave expansion model, similar to that analyzed in [6]. Simulating the propagation of NWB coverage as an expanding wave front (expanding away from S) allows each node v to determine a unique node u from which it initially receives a copy of NWB packet *nwb_id*. Upon receipt of this initial copy, the LENWB algorithm as specified in Section III is computed at v.

Table I

Scenario	No. Nodes	No. Trials	Network Dimensions
1	25	120	530m-by-530m
2	50	80	750m-by-750m
3	100	40	1060m-by-1060m
4	200	20	1500m-by-1500m
5	400	20	2120m-by-2120m

For comparative purposes, a greedy algorithm that utilized complete topology information was considered in along with the LENWB algorithm. Initially, the greedy algorithm begins with X consisting only of S. In each iteration of the procedure, the node in $C(X)$ whose addition to X will maximize the cardinality of the resulting set of covered nodes is added to the set of existing X. If the resulting $C(X) = V$, the algorithm stops. Otherwise, it proceeds for at least one more iteration.

It is easy to contrive network topologies for which such a greedy solution is not optimal. However, for many network topologies a greedy solution is optimal. Further, it is shown in [19] that a greedy heuristic shares the same upper bound ($|X| \leq |V|+1-(2\times|E|+1)^{0.5}$) as that of optimal solution to the problem of finding a dominating set of minimum cardinality. It is also worth noting that the problem of minimizing $|X|$ for NWB is very similar to minimizing the cardinality of a dominating set. In fact, as shown in Appendix C, the decision problem related to obtaining a NWB cover can be restricted to a variation of the dominating set decision problem. Since dominating set decision problem is known to be NP-complete [12], so is the decision problem related to NWB cover. Calculating the set X with minimum cardinality that provides a NWB cover is, therefore, an intractable problem in general.

For each simulation scenario, the transmission radius (R) of each node was varied from 150m to 300m with a step size of 10m in order to study the effect of varying the average degree of the network. Column 3 of Table I indicates the number of trials that were run each value of R, for each scenario. Each trial consisted of generating a connected random graph with a uniform node distribution, and then computing the LENWB and greedy algorithm transmitting node sets needed to perform NWB for a randomly selected node. The range of R chosen for the simulation scenarios provided a wide range of average node degrees (i.e., node densities) to be evaluated. Values of R less than 150m produced random graphs that were too sparse to be connected for scenario 4. Nevertheless, R = 150m allowed for an average node degree as small as 6 even for scenario 4.

Figures 1, 2 and 3 report the performance of the LENWB algorithm and an ideal greedy algorithm for scenarios 1, 3 and 5, respectively. (Figures for the results of scenarios 2 and 4 were omitted due to space considerations. The performance trends for these scenarios, however, are similar to those depicted in Figures 1-3.) In each figure, the curves corresponding to the two algorithms represent the average number of transmitters required to perform NWB as R is varied. The curve corresponding to the average degree represents the average network node degree as R is varied.

Table II reports the average efficiency gain (ε_{avg}) of the LENWB and greedy algorithms for each of the five simulations. The values of ε_{avg} were obtained by calculating the mean ε over the range of R simulated for each scenario.

Table II

Scenario	Mean Degree	LENWB ε_{avg}	Greedy ε_{avg}
1	9.110	3.940	4.840
2	10.67	3.564	4.991
3	11.85	3.386	5.233
4	12.69	3.261	5.406
5	13.29	3.176	5.588

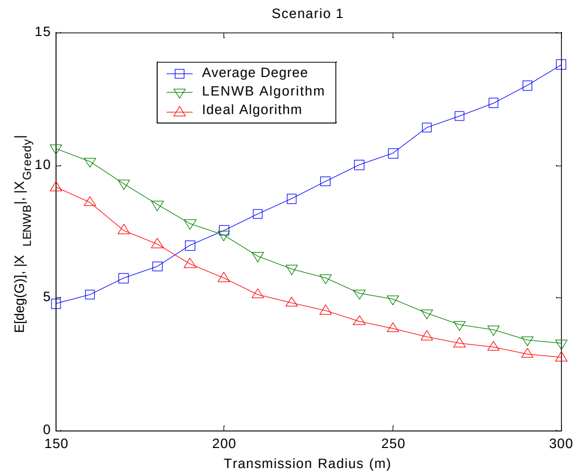

Figure 1: 25 nodes, 530m-by-530m

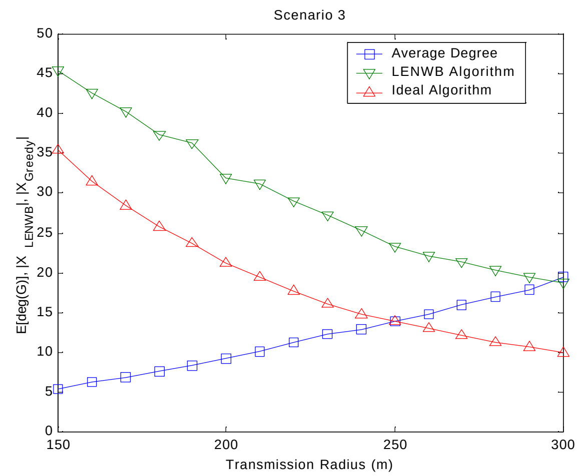

Figure 2: 100 nodes, 1060m-by1060m

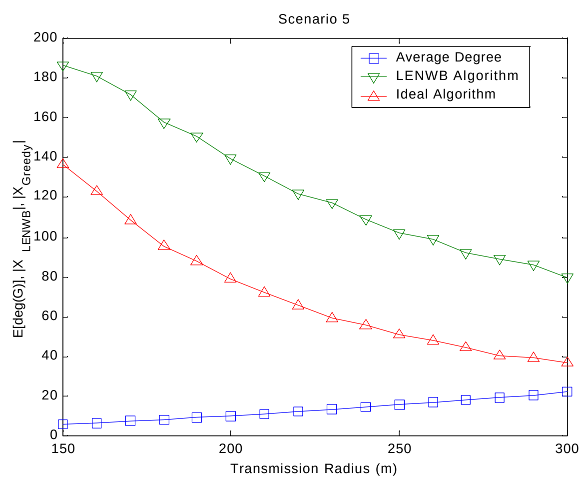

Figure 3: 400 nodes, 2120m-by-2120m

V. DISCUSSION OF RESULTS

Based on Figures 1-3, it is apparent that as the transmission radius increases, so does the average node degree of the network. Increasing the average node degree increases the average number of nodes covered by a single node's transmission. The LENWB and greedy algorithms use local and global topology information, respectively, to block transmission by nodes with relatively low degree whose neighbors will be covered by transmissions from nodes with higher degree. This allows both algorithms to more efficiently disseminate NWB packets throughout the network because high average node degree results in large coverage, on average, by each transmitting node[2]. The expansive coverage by a few strategically elected transmitters means that many nodes need not propagate a given NWB packet. Hence, the slopes of the curves corresponding to the NWB transmitter counts are negative.

From Table II, it is evident that the LENWB algorithm requires substantially fewer transmissions than flooding, by an average factor of 3 to 4. Further, the LENWB approach comes within a factor of 1.2 to 1.8 of the savings provided by a greedy algorithm that exploits complete topology knowledge. LENWB, therefore, represents a realistic solution to the broadcast storm problem discussed in [7].

The capability to exploit modest topology information to afford efficient NWB makes the LENWB algorithm very well suited for on-demand routing protocols. The LENWB algorithm provides substantial savings, as compared to flooding, even though the LENWB algorithm requires only that nodes maintain topology knowledge of other nodes lying within a two-hop radius. However, the AODV and DSR protocols do not proactively maintain two-hop topology information. Nodes running AODV [1] maintain only single hop topology knowledge via Hello messages while nodes running DSR [2] are completely reactive and acquire route information only through tapping and snooping of incident packets or by initiating route queries.

Clearly, the LENWB algorithm must be paired with a hybrid on-demand routing protocol. That is, a routing protocol where nodes proactively maintain two-hop topology knowledge and employ some form of reactive route discovery procedure to learn routes to nodes outside the two-hop proactive routing radius. Thus, the AODV [1] and DSR [2] protocols as currently defined cannot employ LENWB. However, it should be feasible to retrofit either AODV or DSR with an enhanced Hello protocol where the neighbor and degree information required by LENWB are conveyed in each Hello packet. A node running the hybrid version of AODV or DSR will route via topology information learned from the Hello protocol to nodes up to two hops distant from itself. When a route to a node situated outside of this two-hop proactive routing radius is required the node will invoke route discovery and broadcast a RREQ packet to its neighbors. To efficiently implement NWB of this RREQ packet, all nodes apply the LENWB algorithm to determine whether they should propagate a copy of the packet.

An enhanced version of AODV or DSR that incorporates the LENWB algorithm as part of the route discovery process will yield a significant reduction in RREQ packet overhead. In the case of DSR, whenever a propagating RREQ packet is issued, it is possible that all nodes (except for the target node, *T*, if reached) within the *hop_limit* radius of *S* will broadcast a copy of it. With LENWB operating at each network node, the fraction of nodes that actually propagate a copy of the packet is substantially reduced. Similarly, in the case of AODV where an incrementally expanding ring search may be employed in the route discovery procedure, it is possible that all nodes within the radius of the current ring search will transmit a copy of the associated RREQ packet. Here again, LENWB will reduce the fraction of nodes serving as transmitters to disseminate a RREQ packet. Thus, for both protocols, the LENWB algorithm will serve as a blocking process to reduce redundant transmissions.

It is relatively straightforward to see how the LENWB algorithm can also be applied to the ZRP. For example, in a network where nodes are running ZRP with a 2-hop proactive routing radius, LENWB can be directly applied for NWB by simply replacing the bordercast routing protocol. This will potentially save redundant transmissions that would otherwise occur during route discovery, when peripheral nodes of a bordercasting source do not have a route to *T* and each initiates its own bordercast session. Further, the LENWB algorithm can also be applied when the proactive routing radius of the ZRP implementation exceeds two hops. The LENWB algorithm still potentially eliminates redundant bordercast sessions that might otherwise be initiated at peripheral nodes.

[2] It is curious to note that the performance of the greedy algorithm is approximately $2\times|V|\div deg(G)$, where $deg(G) \equiv$ the average degree of the network nodes.

However, since the LENWB algorithm considers only two-hop topology information, the *h*-hop topology information available in ZRP implementations with an *h*-hop proactive routing radius ($h > 2$) would not be fully exploited. For such network scenarios, the basic LENWB algorithm should be modified to fully exploit *h*-hop ($h > 2$) topology information that is available.

Another observation on the data reported in Table II is that the performance of the LENWB algorithm, as compared with the greedy algorithm, diminishes as the network node count increases. Part of the comparative disadvantage with respect to the greedy algorithm is due to growth in ε_{avg} that is apparently inherent in the greedy algorithm as the average degree of a network increases, as evidenced by second and fourth columns of Table II. Further, LENWB is very competitive with respect to the greedy approach when the average network node degree is small, even for scenario 5 where the node count was 400.

The LENWB algorithm provides a diminishing ε_{avg} as the node count increases, as indicated by the third column of Table II. Presumably, this is due to uncovered nodes in $N(v)$ that receive a copy of the NWB packet via a path of forwarding nodes that is not visible to *v*. A node *v*, running the LENWB algorithm, will therefore not be cognizant of such a circulation of copies of the NWB packet. Thus, upon erroneously determining that one or more nodes in $N(v)$ are uncovered, *v* will transmit a redundant copy of the NWB packet. This blind side coverage of a particular region of the network topology is not as likely in networks with small diameters. In such a network, a significant percentage of the network nodes are situated on the perimeter of a network. Nodes on the perimeter can effectively be reached from only one direction, so to speak. That is, a perimeter node is likely to be reached via transmissions that come from the center of the network topology, although it is possible for a NWB to propagate around the perimeter of the network and arrive at a given node from two directions. Such scenarios are improbable and hence the improved performance in networks with smaller diameter.

Lastly, it is worth noting that the *rate* at which the efficiency gain of the LENWB algorithm is reduced as the network node count increases, is monotonically decreasing as indicated by the third column of Table II. This suggests that the LENWB efficiency gain may actually level off at a value well above unity. In fact, extrapolating the limited available data suggests that if the node count was doubled indefinitely to infinity, the ε value would level off at a value of about 3, in the limit. This conclusion is supported by the data plotted in Fig. 4. Thus, the LENWB algorithm appears to be not only an efficient NWB approach for the scenarios considered, but also a scalable algorithm with respect to increasing node count.

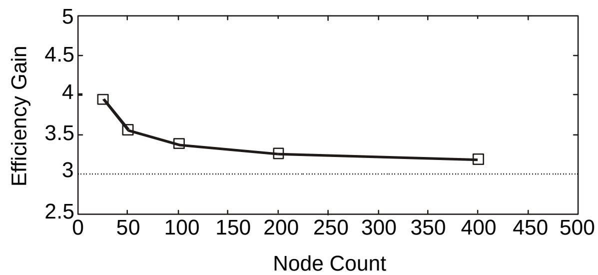

Figure 4: Trend of LENWB ε_{avg}

VI. CONCLUSIONS AND FUTURE WORK

In terms of further evaluation of LENWB performance, assessing the reliability of the LENWB algorithm under conditions where NWB packet transmissions are not assumed to be reliable is the next logical simulation study. Although route discovery in on-demand routing protocols can tolerate a certain amount of unreliability in the RREQ packet propagation, the lack of reliability should not become so great that an excessive number of route discovery attempts need to be made in order for *S* to learn a route to *T*. Additionally, the impact of node mobility while NWB packets are propagating should be considered.

Retrofitting the LENWB algorithm to AODV, DSR and ZRP, and subsequent evaluation of these protocols performance with LENWB managing RREQ propagation represents another direction for future work. Further, the LENWB algorithm may need to be extended to fully exploit the additional topology information provided by ZRP when the proactive routing radius of its implementation is greater than 2 hops.

The version of the LENWB algorithm, as described in Section III and as implemented Section IV, may be fine-tuned with additional heuristics. For example, when a node *v* that is running the LENWB algorithm receives a copy of the NWB packet from more than one neighbor, the coverage of multiple neighboring forwarding nodes will likely be greater than that of a single neighboring forwarding node. This will increase the likelihood that *v* need not propagate the packet. Study of additional heuristics might also involve applying some of the query containment measures considered in [7].

In summary, the application of the LENWB algorithm for NWB in MANETs shows great promise as an efficient alternative to flooding. In particular, adapting LENWB to on-demand routing protocols such as AODV and DSR will allow it to see frequent use for route discovery. The reduction in RREQ packet overhead yielded by LENWB will improve the scalability of these protocols.

Appendix

A. Pseudo Code for Implementation of Steps 3-5

```
P = ∅;
Q = ∅;
For each node c ∈ C(u) do
   If PC(c,v) is TRUE
      If c ∈ N(v)
         Q = Q ∪ {c}
      Else
         P = P ∪ {c};
      EndIf
   EndIf
EndFor
While (N(v) – C(U) ≠ ∅) & ((P ∪ Q) ≠ ∅) do
   If Q ≠ ∅
      q = min(Q);
      C(U) = C(U) ∪ N(q);
      U = U ∪ {q};
      For each node c ∈ N(q) do
         If (PC(c,v) is TRUE) & (c ∉ U)
            If c ∈ N(v)
               Q = Q ∪ {c}
            Else
               P = P ∪ {c}
            EndIf
         EndIf
      EndFor
      Q = Q – {q};
   Else
      p = min(P);
      C(U) = C(U) ∪ N(p);
      U = U ∪ {p};
      For each node c ∈ N(p) do
         If (PC(c,v) is TRUE) & (c ∉ U) & (c ∈ N(v))
            Q = Q ∪ {c};
         EndIf
      EndFor
      P = P – {c};
   EndIf
EndWhile
If N(v) – C(U) ≠ ∅
   Queue packet nwb_id for transmission
Else
   Discard packet nwb_id
Endif
```

B. Proof of LENWB Reliability

To prove that the LENWB algorithm successfully disseminates a NWB packet throughout a connected network, under the condition of zero packet collisions and zero node mobility, two possible scenarios are considered. One scenario is where the algorithm yields no savings over flooding. In this case, $|V|$ nodes will transmit a copy of the NWB packet. That is, every node will transmit a copy of it. The fact that every node transmitted a copy of the packet means that every node must have received a copy of it.

In the second scenario, the LENWB algorithm yields a set of transmitting nodes (X), where $|X| < |V|$. Since $|X| < |V|$, it is possible that an arbitrary X may yield a coverage set $C(X)$, such that $|C(X)| < |V|$, i.e., $C(X)^c = V – C(X) \neq \varnothing$. Suppose that the LENWB algorithm has yielded such an X. Since the original network graph, $G = (V,E)$, is assumed connected, each connected induced subgraph consisting of nodes belonging to $C(X)^c$ must have at least one neighbor, say v, satisfying the following three requirements: (R1) $v \in C(X)$, (R2) $v \in X^c$ and (R3) $N(v) \cap C(X)^c \neq \varnothing$. This means that v did not propagate a copy of the NWB packet even though there were neighboring nodes of v that were not in $C(X)$. In order for there to exist a set X for which the supposition $|C(X)| < |V|$ is true, all three requirements R1, R2 and R3 must hold at a node v for at least one of the following four possible conditions. For cases (ii) through (iv), u will designate the node from whom v originally received the RREQ packet.[3]

i) $v \in C(X)^c$.
   This contradicts the supposition because requirement R1 is not satisfied.

ii) $N(v) \subseteq C(u)$.
   $v \in C(u) \subseteq C(X)$ which satisfies R1. In accordance with step 2 of the LENWB algorithm running at v will terminate and v will not propagate a copy of the NWB packet, which satisfies requirement R2. However, since $N(v) \subseteq C(u) \subseteq C(X)$, $N(v) \cap C(X)^c =$

[3] It will be assumed here that all nodes whose transmissions will cover $N(v)$, received their original copy of the NWB packet via a path π from a single node u. The more general scenario where it is possible for another node $w \neq u$ to serve as an entry node for a NWB packet to reach $N(v)$, as in $w \rightarrow x \rightarrow N(v)$, will not be considered here for the sake of brevity. The proof for the general case, however, would simply extend the logic given here which assumes no such w.

$\varnothing$, which contradicts the supposition because requirement R3 is not satisfied.

iii) $v \in C(u)$, $N(v)$ is not covered by $C(u)$ but $N(v) \subseteq C(\{u\} \cup P \cup Q)$.

$v \in C(u) \subseteq C(X)$ which satisfies R1. By step 5, node v will defer transmission to some suitable set $\{u\} \cup P' \cup Q'$, where $P' \subseteq P$ and $Q' \subseteq Q$, which satisfies requirement R2. However, since $N(v) \subseteq C(\{u\} \cup P' \cup Q')$, $N(v) \cap C(X)^c = \varnothing$, which contradicts the supposition because requirement R3 is not satisfied.

iv) $v \in C(u)$, $N(v)$ is not covered by $C(\{u\} \cup P \cup Q)$.

$v \in C(u) \subseteq C(X)$ which satisfies R1. Since $C(\{u\} \cup P \cup Q)$ does not cover $N(v)$, the algorithm terminates in step 5 with node v broadcasting a copy of the NWB packet to cover $N(v)$. Thus, $v \in X$ and the supposition is contradicted because requirement R2 is not satisfied. Further, since v has transmitted and, therefore, $N(v) \cap C(X)^c = \varnothing$, requirement R3 of the supposition is not satisfied, either.

Contradiction of the supposition under conditions (i), (ii) and (iv) requires no further discussion. Contradiction of the supposition under condition (iii), on the other hand, requires further analysis to show that a sufficient set of nodes in P and/or Q, will indeed transmit to cover $N(v)$.

Revisiting the details of step 4 and 5 of the LENWB algorithm as pertaining to condition (iii), v defers transmission to a set of nodes $Z = P' \cup Q'$, where $P' \subseteq P$ and $Q' \subseteq Q$. By steps 4 and 5 of the algorithm, if all nodes in Z transmit then $N(v)$ will be covered. Clearly, if such a set Z does transmit, then the original supposition will be contradicted because there will no overlap between $N(v)$ and $C(X)^c$. However, it remains to be in shown that $N(v)$ will in fact be covered by Z or any other set of nodes. For convenience and without loss of generality, it is assumed that the LENWB algorithm calculates a theoretical set of transmitters Z to cover $N(v)$, such that $|Z|$ is minimal. Therefore, if any single member of Z defers transmission, $N(v) \cap C(X)^c \neq \varnothing$.

First it is noted that for any $z \in Z$, $N(z) \cap N(v) \neq \varnothing$, and that at least one node in $N(z) \cap N(v)$ belongs to $C(X)^c$. Thus if z does not transmit then neither $N(z)$ or $N(v)$ will be covered. It is, therefore, sufficient to show that if $N(z)$ is covered, then so is $N(v)$. Letting $|Z| = 1$ and $Z = \{z\} \subseteq C(u)$ be the single additional node whose broadcast of the NWB packet will cover $N(v)$, then $N(v)$ is covered if z transmits and partly uncovered, otherwise. Since $z \in Z$ for the algorithm running at v, $PC(z,v)$ holds. Node z is aware of this and, therefore, will not defer transmission to v. Clearly, if via the instance of the LENWB algorithm running z, z determines that it should transmit then $N(v)$ is covered as required. If, however, z determines that it does not need to transmit, then by steps 4 and 5 of the algorithm it must be due to z finding a set of nodes to defer to that will cover $N(z)$. Since $(N(v) \cap C(X)^c) \subseteq N(z)$, $N(v)$ will also be covered for cases where z decides to defer transmission to other nodes.

Considering now a node v where $|Z| = 2$, $Z = \{z_1,z_2\}$. Here, it is possible that $Z \subseteq C(u)$, or $z_1 \in C(u)$ and $z_2 \in C(\{u\} \cup \{z_1\})$, or $z_2 \in C(u)$ and $z_1 \in C(\{u\} \cup \{z_2\})$. Again, for the sake of brevity, only the scenario of $Z \subseteq C(u)$ is considered as the other two scenarios can be analyzed with similar reasoning. Assuming without loss of generality that $PC(z_1,z_2)$ happens to hold, then by the logic given already for the case where $|Z| = 1$, z_1 will transmit to cover $N(z_1)$ and part of $N(v)$. Now, the transmission by z_1 may also cover part of $N(z_2)$ but not all of it because $N(z_2) \cap N(v) \neq \varnothing$. Thus, again by logic equivalent to that used by z_1, z_2 will also determine that it should transmit and cover $N(z_2)$ and the previously uncovered portion of $N(v)$. The argument employed here is extended recursively for cases where $|Z| > 2$.

Therefore, if by steps 4 or 5 of the LENWB algorithm a node v determines that there exists a $Z = P' \cup Q'$, respectively, then v is assured that $N(v)$ will covered by other transmitting nodes. This validates the contradiction of the supposition under condition (iii). Finally, since the supposition does not hold under any of the four possible conditions, the supposition has been contradicted and the reliability NWB by the LENWB algorithm has been proven.

C. *Proof That NWB Decision Problem Is NP-Complete*

Problem: NWB COVER
Instance: Graph $G = (V,E)$, positive integer $k \leq |V|$ and source node $S \in V$.
Question: Is there for G a connected dominating set of size k or less that contains S?

It is evident that the problem NWB COVER belongs to the class NP, because a non-deterministic algorithm can guess a set of vertices $V' \subseteq V$ and verify in polynomial time whether both of the following two conditions are true:

a) The induced subgraph formed by the vertices in V' is in V and is connected.

b) V' is a dominating set with cardinality less than or equal to k.

Now, to show that NWB COVER is NP-hard, the problem is reduced to a variation of the DOMINATING SET problem. It is known that the DOMINATING SET problem is NP-complete and that its variation where the subset $V' \subseteq V$ is required to be connected is also NP-complete [12]. Thus, all that remains to be shown is that with the additional constraint of $S \in V'$, the problem is still NP-complete. This can be easily be shown by restricting the problem NWB COVER to the DOMINATING SET problem by considering graphs $G = (V,E)$ where removal of the node S will leave the graph disconnected (i.e., S is a cutvertex of G). Since removal of S will leave G disconnected, omission of S from V' will also leave V' disconnected, and therefore, violate the condition of a connected dominating set. Finally, since it has been shown that it is possible to restrict the NWB COVER problem to the DOMINATING SET problem (by adding a restriction to G), and the DOMINATING SET problem is known to be NP-complete, the NWB COVER problem is also NP-complete.

References

[1] C. E. Perkins, E. M. Moyer and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing", IETF Internet draft, draft-ietf-manet-aodv-05.txt, 10 March 2000.

[2] J. Broch, D. B. Johnson and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc networks", IETF Internet draft, draft-ietf-manet-dsr-03.txt, 22 October 1999.

[3] R. G. Gallager, P. A. Humblet and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees", *ACM Trans. Program. Lang. and Syst.*, Vol. 5, No. 1, January 1983, pp. 55-77.

[4] I. Chlamtac and S. S. Pinter, "Distributed nodes organization algorithm for channel access in a multihop dynamic radio networks", *IEEE Trans. Comput.*, Vol. C-36, No. 6, June 1987, pp. 728-737.

[5] I. Chlamtac and S. Kutten, "Tree-based broadcasting in multihop radio networks", *IEEE Trans. Comput.*, Vol. C-36, No. 10, October 1987, pp. 1209-1223.

[6] I. Chlamtac and O. Weinstein, "The wave expansion approach to broadcasting in multihop radio networks", *IEEE Trans. Commun.*, Vol. 39, No. 3, March 1991, pp. 426-433.

[7] 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, pp. 151-162.

[8] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE J. Select. Areas Commun.*, Vol. 17, No. 8, August 1999, pp. 1369-1379.

[9] E. Pagnani and G. P. Rossi, "Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks," Mobile Networks and Applications, Vol. 5, No. 4, 1999, pp. 175-192.

[10] M. Jiang, J. Li and Y.C. Tay, "Cluster based routing protocol," IETF Internet draft, draft-ietf-manet-cbrp-spec-01.txt, 14 August 1999.

[11] Z. J. Haas and M. R. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks", IETF Internet draft, draft-ietf-manet-zone-zrp-02.txt, June 1999.

[12] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," 1979.

[13] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.*, Vol. 15, No. 7, September 1997, pp. 1265-1275.

[14]L. Ji and M. S. Corson, "A lightweight adaptive multicast algorithm," *Proc. IEEE Globecom*, 1998, pp. 1036-1042.

[15] A. Segall and B. Awerbuch, "A reliable broadcast protocol," *IEEE Trans. Commun.*, Vol. COM-31, No. 7, July 1983, pp. 896-901.

[16] D. J. Baker, A. Ephremides and J. A. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE J. Select. Areas Commun.*, Vol 2, No. 1, January 1984, pp. 226-237.

[17] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Trans. Commun.*, Vol. COM-29, No. 11, November 1981, pp. 1694-1701.

[18] P. A. Humblet,"A distributed algorithm for minimum weight directed spanning trees," *IEEE Trans. Commun.*, Vol. COM-31, No. 6, June 1983, pp. 756-762.

[19] A. K. Parekh, "Analysis of a greedy heuristic for finding small dominating sets in graphs," *Inform. Process. Lett.*, Vol. 39, Sept. 1991, pp. 237-240.