Robustness Control for Network-Wide Broadcast in Multi-Hop Wireless Networks*

Paul Rogers and Nael Abu-Ghazaleh

Computer Science Department

State University of New York

Binghamton, NY 13902

Email: `progers1@binghamton.edu`, `nael@cs.binghamton.edu`

February 15, 2007

1 Introduction

Mobile Ad hoc networks (MANETs), mesh networks, and multi-hop sensor networks are instances of multi-hop wireless networks where nodes cooperate to forward traffic among each other. Such networks are important whenever infrastructure is unavailable or expensive, and quick deployment is desired. They play an important role in many existing and emerging applications in the military, industry, research and civilian domains.

Network-Wide Broadcast (NWB) algorithms provide a mechanism to deliver information to nodes in a multi-hop network without depending on routing state. This makes them ideal for initial self-configuration or for operation under mobility where the routing state becomes stale. Therefore, NWB is a heavily used primitive at the core of most multi-hop routing [1, 2] and group communication protocols [3, 4]. A common approach to performing NWB is flooding: a process where every node that receives a packet for the first time, rebroadcasts it. Flooding has been shown to be wasteful, especially in dense networks – a problem called *the broadcast storm* [5].

Ni et al, who identified the broadcast storm problem, also proposed several solutions to it based on nodes locally determining whether their rebroadcast is likely to be needed. An alternative, topology-sensitive, approach to the problem attempts to construct a virtual backbone that is tasked with disseminating the broadcast. Such a virtual backbone must form Connected Dominating Sets (CDS) comprising of a connected subset of the nodes that together cover all the nodes in the network; this set can then be tasked with rebroadcasting the NWB packet while all other nodes just receive it significantly reducing the number of retransmissions relative to flooding. We survey these approaches in more detail in Section 3.

While the focus of NWB algorithms has mostly been on reducing the overhead, a different problem occurs that affects the robustness of most NWB protocols. This problem affects NWB protocols that rely on MAC level broadcast operations. This includes virtually all existing NWB algorithms, both flooding based protocols [5] and virtual backbone approaches (e.g., [1, 6, 7]). More specifically, because MAC broadcasts are unreliable, it is possible for rebroadcasts to be lost due to interference or transmission errors. The loss rate can be considerable if high interference exists or if link quality is bad [8, 9]. These losses may lead to the NWB reaching only a subset of the nodes. A particularly damaging example of this problem occurs when the initial transmission of the NWB packet is lost at all receivers (e.g., due to a collision with another packet). In this case the NWB will not reach any nodes.

In Section 5 we present a classification of existing NWB algorithms, focusing on their robustness characteristics. In addition, we analyze the robustness properties of selected algorithms representing different points of the space under different network densities, loss rates, and mobility characteristics; the algorithms are described in Section 4. We show that under high loss rates and in networks that are sparse coverage for existing NWB algorithms, including flooding

*This work is partially supported by AFRL grant FA8750-05-1-0130, and NSF grant CNS-0454298.

and DCB, is poor. In addition, all the evaluated algorithms that reduce redundancy relative to flooding also reduce robustness. More importantly, algorithms that have static rebroadcast responsibilities are more vulnerable to losses than algorithms that take rebroadcast decisions locally. Moreover, algorithms that are topology sensitive perform worse as mobility increases.

The results from the evaluation of NWB algorithms argue for the need of a robustness control component of NWB. Thus, NWB algorithms may be viewed as two components: (1) redundancy control: this is the component which attempts to reduce redundancy while maintaining coverage; and (2) robustness control: this component attempts to recover from lost rebroadcasts and maintain coverage in the face of losses. The vast majority of existing NWB target redundancy control, without considering robustness control.

The next contribution of the paper is to present a classification of the solution space for robustness control in Section 6. More specifically, robustness control may be implemented at the network layer or at the MAC layer. Robustness control may be accomplished by statically introducing redundancy or by dynamically detecting losses and reacting to them. For the dynamic loss-sensitive approaches, we discriminate between approaches that use explicit feedback to detect losses, and those that predict losses based on locally available information. Finally, we discriminate between approaches that are sensitive to the state of the network, and those that are not. This section also presents a classification of existing algorithms that have robustness control components.

In order to evaluate the different approaches to robustness control, we implement representative protocols from different points of the solution space. We study these protocols for a number of different scenarios in Section 7. Finally, in Section 8 we present some concluding remarks.

2 Background – MAC Broadcast

Wireless communication properties make wireless transmissions unreliable for two primary reasons: (1) Channel fading; and (2) Collisions. Channel fading can result in substantial signal power fluctuations at different time scales [10]. This leads to a large number of transmission errors, especially when communicating nodes are far from each other [8, 9].

Neither interference nor transmission occur uniformly across the links of a network. More specifically, interference results in losses in areas closest to the interfering traffic, while transmission errors are affected by the surroundings (which determine the signal fading effects, Doppler effect, as well as the path loss exponent which defines how fast the signal attenuates with distance) and increase with the distance between the sender and the receiver.

In a wireless medium, the channel is non-uniformly shared, giving rise to the well known hidden terminal problem [11]. A hidden terminal is an interfering node out of carrier sense range of the sender, but in interference range with the receiver; such a transmission is not detected by the sender (it is hidden to it), causing a potential collision at the receiver. Techniques such as Carrier Sense Multiple Access and Collision Avoidance reduce collisions but do not eliminate them [12]. Finally, concurrent transmissions can cause collisions especially when correlated transmissions occur within an NWB [5].

To improve reliability in the face of losses, MAC protocols like IEEE 802.11 [13] use retransmission to recover from losses of packets that are not acknowledged. Such an approach cannot be used with broadcast packets because there are a number of receivers. Accordingly, if a broadcast packet is lost due to a collision, **the loss is not detected by the sender and no retransmission is carried out**. This makes MAC level broadcast operations much more susceptible to losses. As a result, NWB operations which rely on MAC level broadcast suffer from losses that can lead to significant loss of coverage: *the NWB robustness problem*.

2.1 Broadcast Robustness

This section outlines the reasons that cause NWB algorithms to lack robustness. Wireless communication channels are unreliable for two primary reasons: (1) Transmission errors: signal propagation effects and interaction with the surrounding environment can cause deep fades in the received signal power (20dB, or 100x changes in signal power are common [14]). These fades can occur in small scales in time and space. This effect leads to a large number of

transmission errors, especially when the two communicating nodes are far from each other; and (2) collisions: the wireless channel is non-uniformly shared, giving rise to the well known hidden terminal problems [11].

Transmission errors and collisions makes MAC level broadcast operations susceptible to losses. The lack of higher level reliability mechanism (e.g., as is present in unicast operations via acknowledgements and retransmission) exacerbates the problem. As a result, NWB operations which rely on MAC level broadcast suffer from loss of coverage.

In addition to losses in the network from various application traffic flows, losses can occur with self-interference. In self-interference, a NWB operation results in two nodes broadcasting at the same time, resulting in collisions at receiving nodes. Since any node can issue a broadcast at any time, this can be problematic in a NWB operation. As a result, many NWB algorithms attempt to cut down on the number of broadcasts needed for a NWB.

Ni et al. studied the effect of all nodes in the network performing a broadcast in [5]. They labeled this phenomenon as the broadcast storm problem. Rebroadcasts carried out by nodes in the network were found to be redundant, as many neighbors had the broadcasted information already. Furthermore, the additional rebroadcasts caused contention in the network to go up, as the broadcasts interfered with each other. Lastly, since there is no backoff mechanism or reliability mechanism built into MAC level broadcasts, the number of collisions in the network went up.

3 Overview and Classification of NWB Algorithms

In this section, we overview NWB approaches and classify them with an emphasis on reliability. The primary characteristics we are interested in are: (1) Overhead; (2) resilience to mobility; and (3) robustness to losses.

3.1 Topology-Ambivalent (Flooding) Approaches

Figure 1: Flooding Example

Flooding is an approach to NWB that operates as follows: every node rebroadcasts a flood packet the first time it receives it (Figure 1 shows an example). Because no topology knowledge is needed, flooding and algorithms derived from it are *resilient to mobility* and do not require an on-going overhead to discover neighbors/topology. Flooding is thought to be resilient to MAC losses due to the high redundancy generally present. However, this may not be true: in low density areas of networks, the available redundancy is low. Moreover, the amount of available redundancy is fixed – it may be too high for some cases and too low for others.

Flooding is a brute force approach that has high overhead, especially in dense networks [5]. Specifically, it leads to: (1) High redundancy in node coverage with many useless rebroadcasts; (2) High number of collisions: with all nodes rebroadcasting, collisions due to concurrent retransmission can occur; (3) High contention for the use of the medium, leading to delays in delivering the NWB. Eliminating the redundancy inherent in flooding and alleviating the above effects are the primary motivations behind most NWB algorithms.

Figure 2: Optimized Flooding Example

Using a geometric argument and ideal unit-disc transmission assumptions, Ni et al [5] show that at most 61% of the rebroadcast coverage area does not overlap with the original transmission; on average this value is 41%. Furthermore, as additional rebroadcasts are overhead, the average non-overlapping area drops rapidly. As a result of this observation, they suggest having nodes determine locally whether their rebroadcast is likely to be redundant, and eliminate it if it is so (an example is shown in Figure 2, where nodes *D* and *E* do not rebroadcast the packet, with no loss of coverage).

In the same paper, several criteria for deciding whether to rebroadcast or not are investigated, including:

1. Probability based: probabilistically a node decides whether or not to rebroadcast. This approach blindly eliminates rebroadcasts—it may not eliminate redundant broadcasts, or it may eliminate critical ones;
2. Counter based: if a certain number of previous rebroadcasts were heard, an additional one is probably not needed based on the geometric argument they developed. Thus, if the threshold of overhead transmissions is reached, the packet is not rebroadcast;
3. Distance based: assuming that the distance of the node from whom we receive a rebroadcast can be determined, the following criterion can be used to decide whether to eliminate a packet or not. If the distance exceeds a certain threshold, the non-overlapping area is significant and the packet can be rebroadcast; otherwise the packet is not rebroadcast.
4. Location based: in this approach, a rebroadcasting node includes its location on the packet. As a node receives rebroadcast packets, it can compute the additional area that would be covered by its own rebroadcast. If this area exceeds a threshold, the packet is rebroadcast, otherwise it is dropped.

The discrimination between needed and redundant algorithms improves in the order of listing of the criteria. However, the amount of needed information also increases. Further, in particular, the distance based and location based algorithms are susceptible to the fact that their ideal premises do not hold in realistic environments. In general, these approaches are successful in reducing the overhead of NWBs relative to flooding. Like flooding, they are resilient to mobility. Because each node locally determines whether its rebroadcast is likely to be needed, the approach dynamically adapts to transmission losses.

The effect of the threshold in the above algorithms is investigated. However, it is likely that a single threshold will not work for all networks, or for all areas of a network. As a result, Tseng et al [15] investigate the effect of adaptively adjusting the rebroadcast thresholds based on observed behavior. For example, in their adaptive counter scheme (where a rebroadcast is suppressed if a certain number of rebroadcasts of the same packet are overhead), the counter value is initialized to 1, and adjusted up based on observed rebroadcasts. The approach significantly improves on fixed threshold performance.

Scott and Yasinsac [16] propose a solution to dynamically adjust the probability of transmitting a broadcast message. This work is built on the gossiping work of Haas et al [17]. The solution attempts to measure node density

dynamically with a ping mechanism, and use that data to transmit the broadcast packet only if significant area and nodes will be covered.

3.2 Topology Sensitive Approaches

Figure 3: CDS Example

These approaches build a virtual backbone that covers all nodes in the network. Ideally, the backbone forms a *Connected Dominating Set (CDS) of the topology graph*. A dominating set in a graph is a set of nodes where every node in the graph is a member of the dominating set or a direct neighbor of a member. In other words, all nodes in the network are one-hop reachable from the CDS. Thus, if only the dominating nodes rebroadcast NWB packets, all vertices of the graph are covered. An example CDS is shown in Figure 3; in this case, only nodes *A* and *B* transmit and all nodes are covered by the two transmissions.

3.2.1 CDS Construction

The Minimum Connected Dominating Set (MCDS) problem has been shown to be NP-complete for unit disk graphs¹, by mapping the NP-hard set cover problem to it [18]. Das et al. propose a centralized heuristic algorithm for deriving an approximate MCDS [19]. However, a centralized approach is susceptible to mobility; changes in topology require global recalculation of the CDS.

More efficient algorithms, based on minimum spanning trees [6], graph-coloring [7], and clustering [5, 20, 21] have been proposed. However, these are also susceptible to mobility because dynamic changes in topology can require significant recomputation of the CDS. A localized CDS computation approach was proposed by Wu and Li [22]. In this approach, nodes track 2-hop neighbor information. A node selects itself to be part of the CDS using a simple rule: if it has two neighbors that are not directly connected it is a member. This rule results in a large CDS. As a result, Wu and Li suggest two rules for pruning the CDS whereby a node whose neighbors are fully covered by another node or a combination of two nodes is pruned from the CDS. Because it only requires information that is localized, this algorithm is better equipped to maintain the CDS in the presence of mobility.

3.2.2 Properties of Topology Aware Algorithms

In order to build a virtual backbone, nodes exchange information, about neighbors; this is an ongoing process. As a result, topology sensitive approaches *degrade (in terms of overhead and/or coverage) with increasing mobility* since the neighborhood information becomes stale more quickly.

¹These are graphs where the neighbors of a vertex are the vertices contained by a disc of radius 1 around the vertex. Unit-disc graphs can represent the topology wireless networks with ideal omni-directional antennas.

In terms of vulnerability to losses, an important distinction between topology sensitive approaches is whether the forwarding responsibilities are statically or dynamically determined. In static approaches, the topology information is used to statically determine forwarding responsibilities. As a result, this approach becomes especially vulnerable to losses: if a loss of a packet to a node with forwarding responsibilities occurs, the remainder of the backbone reachable through it and the nodes they cover will not receive the NWB. In dynamic approaches [23, 24], each node locally determines whether it needs to be part of the CDS based on already heard transmissions. This makes them significantly more resilient to losses. For example, in the Scalable Broadcast Algorithm [24], each node tracks its two hop neighbors. This enables a node A to determine which of its neighbors a node B covered by checking the two hop neighbor information without requiring B to transmit its neighbor list.

In summary, NWB approaches assess forwarding responsibilities either in a topology sensitive way or based on local estimates of the rebroadcast importance. Topology sensitive protocols can provide more optimal NWBs from an overhead perspective, but require on-going overhead to exchange neighborhood information and are more susceptible to mobility. Within the topology sensitive model, another important classification is whether the forwarding responsibilities are assigned statically or dynamically at each forwarding node. Static responsibilities allow more optimized forwarding, but are more susceptible to loss of coverage due to transmission losses.

All the approaches described thus far focus on reducing the redundancy of NWB. However, the robustness of NWB has recently started to receive attention. However, we delay discussion of these approaches to Section 6 where the robustness control solution space is described.

4 Representative Algorithms

In this section, we overview representative algorithms from different points in the solution space. These algorithms will later be used in characterizing the performance of the different approaches. In addition to flooding, the following NWB algorithms were evaluated, in order to provide representatives of the different classes of solutions:

4.1 Location Based Algorithm (LBA)

In this algorithm, a node includes its location in the rebroadcasted packet. Upon reception of a broadcasted packet, the receiving node notes its current location in comparison with the location of the sending node, obtained from the packet header [5]. This information is then used to calculate the additional area which would be covered with an additional broadcast. Based on this information, a node decides whether its rebroadcast provides sufficient coverage to be worth sending. LBA is a dynamic flooding based approach (not topology sensitive). Flooding typically leads to higher overhead in dense networks. By including location information in each packet, the overhead is not raised quite as high as a traditional flood. We describe LBA in more detail by outlining its operation using pseudo-code.

```
When Node A receives a broadcast P from Node B:
    Distance := Distance(B, A);
    If Distance > DISTANCE_THRESHOLD Then
        Rebroadcast(P);
End If;
```

4.2 Ad Hoc Broadcast Protocol (AHBP)

In the Ad Hoc Broadcast protocol [25], nodes track two hop neighbor information via ‘HELLO’ packets and use this information to explicitly select a set of 1-hop neighbors to rebroadcast the packet such that all the 2-hop neighbors are covered. Thus, it is a static CDS based algorithm. Only nodes marked as Broadcast Relay Gateway (BRG) nodes perform a broadcast. The node selection algorithm of AHBP is as follows:

1. Locate all 2-hop nodes that can only be reached by a single 1-hop node. Mark those 1-hop nodes as BRG members.
2. Calculate the set of nodes that will be covered by the BRG nodes.

3. For all remaining 1-hop neighbors, find the neighbor that will cover the most uncovered 2-hop neighbors, and mark it as a BRG node.
4. Repeat steps 2 and 3 until all 2-hop neighbors are covered.

Within a NWB packet, the set of BRG neighbors is kept in the packet header. As a result, upstream neighbors calculate which downstream neighbors will rebroadcast the packet, making this a static CDS-based approach. Static CDS algorithms can suffer greatly if a node with forwarding responsibilities does not receive a packet. In such a situation, the propagation of the NWB is halted, and the NWB is often worthless.

When a BRG node receives a broadcast, it uses its current 2-hop neighbor information to calculate which 1-hop neighbors received the transmission. These neighbors are then removed from the neighbor set used in the BRG selection algorithm described above.

Since AHBP is CDS based, its overhead is significantly smaller than flooding based algorithms. However, since it is topology sensitive, it is susceptible to mobility. Furthermore, since it is static, its robustness is low.

4.3 Scalable Broadcast Algorithm (SBA)

The Scalable Broadcast Algorithm (SBA) [24] is essentially a dynamic version of AHBP where nodes locally determine if all their 2-hop neighbors have been covered based on overhead broadcasts. The 2-hop neighbor information is obtained via 'HELLO' packets.

If node A sends a packet, and it is received at node B, node B calculates if it has neighbors that are not covered by node B. If there are uncovered neighbors, it will retransmit the packet. The algorithm is described in more detail using pseudo-code.

```

Let T1 be the SBA Timer
Let L be the list of nodes that sent a particular broadcast
Let Nx be the neighbor list for node X
Let U be the list of uncovered nodes

When a source sends a packet, it is always broadcast.

When a node A receives a broadcast from node S:

If ({Na} > {Ns}) Then
  If T1<P> Then // If a timer exists for this packet
    T1<P>.add(P.source); // Add the originator of this broadcast
  Else
    T1 := new Timer(P, {Na}-{Ns});
    T1.start();
  End If;
End If;

T1.expire() {
  If ({U} > {N{L}}) Then
    Rebroadcast(P);
  End If;
}

```

Dynamic CDS algorithms adapt to losses far better than their static CDS counterparts. This resilience comes at a cost of higher forwarding costs than the static CDS approaches, but is more resilient to poor network conditions.

4.4 Double-Covered Broadcast (DCB)

DCB [26] is a static topology-sensitive CDS-based algorithm with fixed built in redundancy. Specifically, the CDS is built such that it provides double-coverage for every node. The intuition is that even if a single transmission is not received, the other is likely to enable recovery from this loss. Neighbor information is tracked via ‘HELLO’ packets. A node wishing to send a NWB will follow the following algorithm:

1. Select a subset of 1-hop neighbors that cover all 2-hop nodes. Each of these neighbors must be covered by at least two transmissions, one from the sending node, and one from the 1-hop neighbor set. The nodes selected serve as forwarding nodes.
2. Each forwarding node follows the previous step and broadcasts the packet.
3. The transmission of the forwarding nodes serves as an acknowledgement for the upstream sending nodes. If a sender does not detect rebroadcasts from all forwarding nodes, it recalculates the forwarding node set, and repeats the process.

While this algorithm has built in redundancy, this redundancy is fixed: it may be insufficient in some situations (causing poor reliability), or unneeded in other situations, causing unnecessary addition retransmissions. Furthermore, when recalculating the forwarding node set, the latency of the NWB is increased, as a node must wait for a series of ‘HELLO’ messages to be sent.

5 Characterizing NWB Unreliability

In this section, the effect of the MAC broadcast unreliability on the performance of NWB algorithms is characterized to demonstrate their lack of robustness. NWB coverage is affected primarily by two factors: the redundancy of the NWB operation and the probability of MAC transmission loss. A dense network allows some NWB algorithms to use redundant paths providing loss-tolerance without losing coverage. This available redundancy is lower in sparse networks, or for algorithms that control redundancy to improve overhead, making NWBs less robust under such conditions.

5.1 Preliminaries and Experimental Setup

The primary performance metrics of interest are: (1) node coverage: percentage of nodes that receive the NWB; (2) overhead: number of retransmissions. Raw overhead is difficult to interpret independently of the network size, and the coverage success. Therefore, *normalized overhead*, defined as the number of retransmissions per receiving node, was tracked. For flooding, normalized overhead is always one regardless of the topology or the coverage success: every receiving node retransmits the packet once. Optimized NWBs target lowering the overhead while maintaining coverage which should result in a normalized overhead smaller than 1.

To simulate losses in the network, two approaches are used. First, packets are probabilistically dropped at receiving nodes; a similar approach to modeling losses was used in other works [26]. While this approach is somewhat unrepresentative (transmission errors are typically not uniform across all links), it provides a controllable approach to varying the loss rate. The second approach involves introducing fading to the network. Fading models allow for more realistic transmission behavior, but vary depending on the distance between nodes. As a result, the best way to judge the effect of fading is by comparing NWB behavior in networks of different densities. Previous work has been done using Constant Bit Rate (CBR) traffic flows in order to increase the level of contention present in the network. These flows often interfere with each other, preventing a consistent level of contention in the network. As a result, data gathered from these studies is not as controlled as what is presented below.

We evaluated the algorithms described in Section 4². All experiments in this paper use the Network Simulator NS-2 [27]. NS-2 is a discrete event simulator with detailed models for the network protocol stack, including the IEEE

²The authors wish to thank Tracy Camp’s group at Colorado Mines for the supplying the code for LBA, SBA, and AHBP. We also wish to thank W. Lou for the NS-2 implementation of DCB

802.11 MAC protocol. NS-2 includes simplified models for wireless propagation. In all scenarios, nodes are randomly deployed with a uniform distribution in a fixed area of 1000 by 1000 meters. The 802.11 MAC implementation in NS-2 was used for experimentation with the default parameters (range of 250 meters). In each experiment, one NWB is generated per node. The NWBs are timed to ensure that successive operations do not interfere. Each data point represents an average of 20 scenarios with different random seeds. While the confidence intervals are not presented in the graphs to avoid clutter, they were tracked; the average is narrowly bound (95% confidence intervals within 1% of the mean).

5.2 Grid Topologies

Grid topologies consist of nodes that are evenly deployed in a two-dimensional grid throughout the simulation area. This configuration may be representative of static pre-planned networks such as mesh network [28]. Grids also provide a level of control to experiments, due to the uniform density present throughout the network. The coverage of flooding for different density grids inside of a 1000m by 1000m area were studied.

Figure 4: Coverage of Flooding (Grid Deployment, Controlled Drop)

Figure 4 plots the coverage achieved by flooding in different densities, using the controlled drop mechanism described earlier. As expected, grids that are more dense achieve a higher level of coverage. This is due to the fact that denser grids provide more redundancy for the flood. In a sparse grid, the loss of a few packets can result in loss of coverage to significant portions of the network.

5.3 Random Topologies

This section describes studies done with nodes that are randomly deployed in a 1000m by 1000m area.

5.3.1 Controlled Drop

Figure 5 charts the coverage obtained for flooding in different densities of networks. Similar to the density study done for grid scenarios (Figure 4), as the density increases, the overall coverage increases. This is because the redundancy built into the network has increased.

Figures 6 and 7 plot the coverage and overhead for various NWB protocols in a sparse 30 node environment. As the loss rate is increased, the overall coverage of the protocols decreases. This is particularly true for CDS-based approaches, where the loss of one packet in the CDS tree can stop the progression of the NWB.

The coverage of flooding and LBA is very similar. However, LBA optimizes the number of rebroadcasts needed based on location, resulting in a slightly lower overhead. SBA, being a dynamic CDS algorithm, adapts to losses in the network, resulting in coverage near flooding. Furthermore, as the loss rate increases in the network, the number

Figure 5: Coverage (Random Deployment, Controlled Drop, Flooding)

Figure 6: Coverage (Random Deployment, Controlled Drop, 30 Nodes)

Figure 7: Overhead (Random Deployment, Controlled Drop, 30 Nodes)

of broadcasts performed by SBA increases as well. This demonstrates the adaptive nature of the SBA protocol. For the static CDS approaches, DCB has a higher coverage level than AHBP. This behavior is expected, as each node is covered twice, as presented earlier.

The controlled drop experiments were repeated in a denser, 60 node environment (Figures 8 and 9). As the network is more dense, the protocols are generally more tolerant to losses in the network, resulting in higher coverage. This is especially true of flooding and LBA. The CDS-based approaches make a sparse network out of a dense network, and therefore lose some of the built in redundancy of a dense network. At high loss levels, the static approaches of AHBP and DCB drop off when compared to the dynamic approach of SBA and the flooding algorithms.

In terms of overhead, since more nodes are covered per broadcast, the overall normalized overhead tends to be less than that of a sparser network. The gain of using LBA over flooding is larger in a dense network, since less nodes rebroadcast per region of the network.

5.3.2 Losses Due to Self-Interference

As nodes propagate a NWB packet, there is a chance that the reception of that packet will coincide with the reception of another packet. If both packets are for the same NWB, those losses are considered to be self-interference. To measure levels of self-interference, the studies from above were repeated with the Two Ray Ground propagation model, with no controlled drop. All resulting losses were due to collisions with other NWB data packets.

Examining Table 1, it can be seen that the flooding-based approaches have a higher level of self-interference than the neighbor knowledge protocols. This is as expected, since the neighbor knowledge protocols attempt to build a smaller set of rebroadcasting nodes. In addition, the dynamic approach of SBA has a higher level of self-interference, due to its higher overhead. AHBP, SBA, and DCB also all try to randomize when a packet is retransmitted

Figure 8: Coverage (Random Deployment, Controlled Drop, 60 Nodes) Figure 9: Overhead (Random Deployment, Controlled Drop, 60 Nodes)

<i>Protocol</i>	<i>30 Nodes</i>	<i>60 Nodes</i>
Flooding	12.021	87.569
Location-based	7.24	69.228
AHBP	1.151	10.065
SBA	1.427	24.961
DCB	0.118	0.82

Table 1: Average number of packet drops per NWB due to collisions

by introducing some jitter. The jitter algorithm used in DCB is significantly more randomized than AHBP and SBA, as very little self contention exists in the protocol.

5.4 Cluster Topologies

Cluster topologies are networks that have groupings of nodes in specific areas, rather than randomly scattered throughout the network. A cluster scenario was studied with 4 clusters equidistant from each other, connected by a cluster of 4 nodes. The resulting shape was that of a “plus”.

5.4.1 Controlled Drop

Figure 10: Coverage (Cluster Deployment, Controlled Drop) Figure 11: Overhead (Cluster Deployment, Controlled Drop)

Figure 10 plots the coverage achieved by the different protocols in the cluster scenarios. The results roughly

mirror the results of the random topologies, with flooding achieving the highest level of coverage. Figure 11 plot the overhead of the different protocols. The results of the cluster experiments have lines that are not as smooth, as a loss of a NWB transmission has a high chance of preventing the rest of the network from receiving the broadcast. The results are somewhat similar to the results found in the 30 node random deployment environment (Figures 6 and 7). This is because a 30 node environment in a 1000m square area is very sparse and loosely connected. The loss of a transmission along one of the links has a high probability of terminating the NWB propagation. As the network density increases, this probability is lowered. By the same logic, if the clusters were made larger, with more nodes, the coverage results would increase proportionally.

5.5 Mobile Topologies

This section presents the coverage and overhead studies for various NWB protocols in a few different mobile environments. Topologies containing nodes within a 1000m by 1000m area were used for these studies.

5.5.1 Random Waypoint

The Random Waypoint Mobility Model is a model commonly used in MANET research. In this mobility model, nodes move in varying directions and speeds, with a pause time between each movement and direction change.

Figure 12: Coverage (RW, Controlled Drop, 30 Nodes)

Figure 13: Overhead (RW, Controlled Drop, 30 Nodes)

Figure 12 shows the tests using the random waypoint model with a pause time of one second between movements, and a maximum speed of 15 meters per second. This figure represents the coverage achieved for different NWB protocols. Similar to the static scenarios, flooding and LBA do better than the neighbor knowledge protocols. This behavior is noticed even more in a mobile environment, because the mobility causes neighbor knowledge to become stale. When Figure 12 is contrasted with the static 30 node results shown previously in Figure 6, the staleness of the neighbor knowledge data can easily be seen. Figure 13 shows the overhead of the protocols in a 30 node environment. The trends seen here are similar to those seen in the static scenarios.

Figures 14 and 15 show the coverage and overhead obtained in a 60 node environment, using the random waypoint model (1 second pause time, 15 meters/sec maximum movement). These results again show that dense networks are more robust and resilient to losses. The overall coverage increases, and the overhead goes down, as more nodes are reached with the broadcasts that are sent.

5.6 Probabilistic Random Walk

The probabilistic version of the random walk model uses a set of probabilities to determine the next position of a node. This model also uses random directions and speeds, but makes use of a matrix in order to determine the node position for the next time slot. Since the movements are probabilistic, the movements of nodes tend to be more realistic than purely random movements [29]. An interval of 0.5 was chosen for these studies.

Figure 14: Coverage (RW, Controlled Drop, 60 Nodes)

Figure 15: Overhead (RW, Controlled Drop, 60 Nodes)

This mobility model was chosen for study since there have been studies that have shown the random waypoint mobility model is not always an accurate representation of mobility [30]. Their studies have shown that the general speed of the network slows over time. Bettstetter and Wagner have also shown [31] that the nodes tend to converge on the center of the network over time, when using the random waypoint model.

Figure 16: Coverage (PRW, Controlled Drop, 30 Nodes)

Figure 17: Overhead (PRW, Controlled Drop, 30 Nodes)

Figure 16 shows the coverage obtained with the probabilistic random walk in a 30 node environment. The dropoff is much sharper, when compared with the random waypoint graphs presented earlier (Figure 12). This is due to the effects mentioned earlier; that the random waypoint mobility model tends to slow down over time and converge towards the middle. As the network converges, and the neighbor information data is less likely to be out of date, the redundancy of the network is increased.

Figure 17 shows the overhead for the probabilistic random walk in a 30 node environment. Again, overall overhead is higher with this model, when compared with the random waypoint data (Figure 13).

As one would expect, the coverage obtained for a 60 node environment is higher than that of the 30 node environment, as can be seen when contrasting Figures 18 and 16.

5.7 Notes

The experiments presented in this section show that in sparse environments and environments under high load, the level of redundancy present in the NWB protocols is often not enough. The protocols can not adapt to losses and therefore are not robust. This weakness needs to be understood and addressed, which we will do in the next sections.

Figure 18: Coverage (PRW, Controlled Drop, 60 Nodes) Figure 19: Overhead (PRW, Controlled Drop, 60 Nodes)

6 Robustness Control – Solution Space

Increasing NWB reliability requires increasing the probability of the reception of the NWB rebroadcast operations. This is especially true *in situations where their loss is likely and the redundancy in the network is low*. The goal of the solutions presented in this paper is to improve the reliability of the NWB, rather than ensure complete reliability; guaranteed reliability requires too much overhead (for example, neighbor discovery and the use of unicast packets).

With solutions for robustness control, NWB algorithms can now be thought of in two parts: (1) Redundancy control: this phase of the algorithm targets reducing the redundancy in the rebroadcasts; and (2) Robustness control: this phase of the algorithm attempts to recover from losses occurring in the transmissions necessary to cover the nodes. The focus of most existing solutions is redundancy control; in contrast, our focus here is on robustness control.

There are several possible approaches for redundancy control which are classified in the remainder of this section. Please note that even though robustness control is discussed separately from the remainder of the NWB and redundancy control aspects of the algorithm, often the most appropriate choice is influenced by the structure of the underlying algorithm. Examples of such inter-dependence will be discussed as different points in the solution space are examined in more detail in the following three chapters.

Approaches to robustness control can be classified along multiple axes, including the following. First, we distinguish solutions according to how they trigger robustness control responses (typically additional rebroadcasts). *Fixed redundancy approaches* incorporate a fixed degree of redundancy in their coverage, beyond the minimum necessary, to leave a safety margin against broadcast losses. In contrast, *loss sensitive approaches*, trigger additional redundancy (typically in the form of additional rebroadcasts) only when losses are detected or predicted.

Among loss-sensitive approaches, there is a number of solutions possible based on the approaches for detecting/predicting losses and how redundancy is built in response. Broadly, losses detection/prediction can be accomplished by *explicit* or *implicit* feedback.

6.1 Explicit Feedback Algorithms

In explicit feedback algorithms, the receivers inform the senders of whether the broadcast was received successfully using an explicit transmission. Explicit feedback solutions face the following problem: if multiple receivers of a MAC broadcast have to provide acknowledgements, collisions occur. However, its possible to randomize/stagger the responses or require feedback from a subset of the receivers only.

With explicit feedback, the granularity of the feedback can be adjusted. The simplest scheme is to provide feedback on every NWB transmission. This has the cost of immediate latency in a NWB operation. Another approach that can be used is to provide feedback about NWB successes is to inform the sending node periodically. This requires nodes to cache previously sent data, in case a failure situation arises. Depending on the granularity and the subset of receivers responding, explicit feedback may also increase latency and load on the network (due to the acknowledgement traffic).

6.2 Implicit Feedback Algorithms

Schemes that use implicit feedback attempt to judge the success of a NWB transmission based on locally observed behavior and without requiring additional control packets to be exchanged. Briefly, this approach hypothesizes that the behavior of the neighbors will be different in the case that they receive a broadcast from the case where they don't. Thus, by observing the behavior of nearby nodes, the loss probability can be predicted. A simple example of such implicit feedback is for a node to observe the channel and see if the NWB packet was retransmitted by neighboring nodes. If enough retransmissions are overhead, the original transmission to the neighbor was likely received successfully.

Implicit feedback solutions can improve their decision by factoring in other criteria that can improve the loss prediction. For example, each node can measure the utilization of the channel; the higher the utilization the higher the probability of a loss. However, this approach is imprecise because the local state at the sender may not match that at the receivers. An additional metric of interest is the expected number of rebroadcasts (a function of density and NWB algorithms) the knowledge of which can help refine the prediction made by the implicit feedback mechanism.

An interesting hybrid approach uses explicit feedback to estimate the probability of success of a broadcast and perhaps the criticality of it; this approach is termed *state-aware*. This approach is similar to the utilization and density estimation discussed above. However, because those estimates are created based on the imprecise source view, they may not be valid at the destination where losses occur. Further, link utilization is only one of the factors that influences the quality of the wireless channel; other factors cannot be assessed without feedback from the receiver [32].

State-awareness can be combined with implicit feedback or fixed redundancy algorithms. Most NWB solutions view the link quality as boolean - links either exist or they do not. The quality of links differs due to many variables, including the distance between the nodes, the power of transmissions between the nodes, and the overall traffic nearby [32]. NWB solutions can leverage this data in order to provide NWB robustness, for example, by building in more redundancy for nodes that are covered only by weak transmissions in a fixed redundancy NWB algorithm.

6.3 Classifying Existing Solutions

Full reliability for NWBs is an expensive operation. Using the 802.11 Medium Access Control (MAC) layer, the only mechanism for a reliable transmission is with unicast packets, which have an optional Request To Send/Clear To Send (RTS/CTS) handshake, followed by the data transmission, followed by an acknowledgement (ACK) packet. If the ACK is not received by the sending node, the process is retried, up to 7 times.

To address NWB robustness, two general approaches are used: (1) MAC broadcast changes: changes to the MAC layer are introduced to make NWB operations more reliable; and (2) NWB algorithm changes: NWB are done with a smaller set of nodes to cut down on overhead.

Tang and Gerla proposed a full reliability MAC protocol to deliver NWB data [33]. This protocol requires all nodes to maintain a neighbor list, a queue of previously sent broadcast packets, and a queue of previously received broadcast packets. Nodes maintain 1-hop neighbor lists by periodically sending out 'HELLO' packets. When performing a NWB, a source node selects a neighbor, and performs a RTS/CTS handshake. When sending the RTS, the packet also contains a piece of data indicating which NWB sequence number is about to be transmitted. The neighboring node replies with a CTS message, containing the last successfully received NWB id. The source node is then responsible for sending all NWB packets between the two values. This process continues in a round-robin fashion throughout the network.

This solution is an example of an explicit feedback algorithm. Their solution has a level of overhead, due to the extra packets that must be exchanged with each node, as well as a high amount of latency, if packets are dropped.

The solutions of Hyper-flooding [34] and Hyper-gossiping [35] are more examples of explicit feedback algorithms. When nodes come into range, 'HELLO' packets are sent, indicating that they are a new node, and that the NWB data should be resent.

Lou and Wu's Double Covered Broadcast (DCB) [26] solution is one that is an implicit feedback solution. Sending nodes keep track of which neighbors rebroadcast the NWB packet. If that retransmission is not heard, it is assumed that the first transmission failed.

Pleisch et al propose the Mistrial protocol [36] as an efficient flooding mechanism. This approach has nodes send out compensation packets periodically. The compensation packets contain versions of previous data packets, and

nodes can deconstruct any missing packets. Since the extra compensation packets are sent without loss notification, it is a fixed redundancy algorithm.

We have also investigated a number of solutions from different points in the solution space. The Selective Additional Rebroadcast (SAR) protocol is a network layer implicit feedback solution. It requires no protocol transmissions to determine whether the packet was received successfully. This solution makes use of packet retransmissions, the contention of the channel, and past SAR successes in order to gauge whether or not to rebroadcast a packet. This work differs from Hyper-gossiping, because it is adapted to many NWB protocols, uses more advanced retransmission criteria, and does not incur the overhead of data caching. This work is different from the additional rebroadcast solution presented in this paper for a number of reasons. The SAR approach uses more sophisticated mechanisms for gossiping, and also builds off of other protocols besides flooding, such as AHBP and SBA. Those protocols have a lower overhead than flooding. Also, the SAR approach does not attempt to cache data for new nodes that come into the network at a later time. The SAR approach is different from DCB in that it attempts to strengthen the chance of a packet reception along the CDS tree, rather than increasing the size of the tree. This causes retransmission to only occur when needed.

Directed Broadcast (DB) is an explicit feedback solution. It requires nodes involved in the DB transmissions to send an ACK packet upon successful packet reception. This feedback is required from only a single neighbor. By using a unicast transmission between nodes, the reliability of unicast is leveraged, while keeping the efficiency of a broadcast. This approach differs from the approach used by Tang and Gerla, described earlier. Transmissions make use of the CDS tree built from neighbor-knowledge protocols, and no significant latency is introduced, as the feedback comes from receiving nodes immediately.

Shadowing-aware CDS is a solution that makes use of the state of the links in a MANET. These links are then used to create a CDS tree that can be used to gauge the chance of a packet reception. Current NWB robustness solutions do not make use of this information.

6.4 Summary

This section overviewed the potential solution space for increasing NWB robustness, along with the intuitive tradeoffs that result from these approaches. Existing solutions were classified in terms of this solution space. In the next section, we evaluate selected solutions from this space.

7 NWB Robustness Solutions

This section presents an evaluation of two approaches for robustness control selected from different points in the solution space. Both mechanisms are independent of the redundancy control NWB aspect of the algorithm and can be used to control its robustness. Selective Additional Rebroadcast is an implicit feedback approach, while Directed Broadcast is an explicit feedback one.

7.1 Selective Additional Rebroadcast

In Selective Additional Rebroadcast (SAR), the base NWB algorithms are modified such that a node may transmit a broadcasted packet a second time [37] based on implicit feedback obtained from the behavior of the network observed by the node. For example, the node can use metrics such as the rebroadcast behavior, the neighboring topology and the nearby network load to determine if a packet loss is likely and if the broadcast was critical. By doing this, coverage robustness is increased, because there is less of a chance that a critical rebroadcasts are lost.

Several criteria can be explored to decide when to rebroadcast a packet an additional time. SAR is designed to provide NWB robustness implicitly. No explicit feedback is used to predict whether a NWB transmission has been lost. This prediction can be based on recently observed behavior such as recent loss rates and the utilization of the channel. In addition, the prediction metrics can take into account the subsequent behavior of nearby nodes (which may differ based on whether the rebroadcast was received or not). We study the following retransmission policies:

1. Probabilistic Solution: a packet is rebroadcast with a fixed probability. In general, this solution is problematic because it does not adapt to the density or loss rates in the network. Therefore, it may result in large increases in the overhead when it is not needed (e.g., in dense areas and/or when interference is low). This solution is included because it provides a reference point that increases redundancy without using any prediction intelligence.
2. Counter-Based Solution: if the node does not hear n other nodes rebroadcast the packet within a certain amount of time, it will rebroadcast it again. This solution is attractive because it naturally adapts to the interference level and density of the network. In a dense/low interference area, a number of rebroadcasts is likely to be received after a node rebroadcasts a packet. But in sparse/high interference areas, this is not the case, and the algorithm rebroadcasts a packet to enhance reliability. First, the behavior of a fixed value of n across the network is studied, followed by the adaptation of the value to reflect the neighboring topology.
3. Adaptive SAR: Rather than having a fixed criteria that applies throughout the network, the appropriate threshold for rebroadcast can be adaptively set at the individual nodes. For example, a node can disable SAR if it determines that the SAR broadcasts have not helped the NWB recently. By doing this, the node can cut down on overhead when the broadcasts are not needed, and re-enable SAR later, in case new nodes have moved into the surrounding area. Alternatively, a node may assess its local neighborhood to compute the expected number of rebroadcasts and set its rebroadcast threshold to match those.

Furthermore, additional improvements to these metrics exist. Examples of these include:

1. Medium Access Control (MAC) Utilization: A node can take into account how busy the medium is, and use that to gauge how likely the loss of a NWB is. This is potentially unrepresentative, since the node is testing how busy the MAC layer is at the source, rather than at the destination.
2. Density: A node can use the number of neighbors it has to determine whether it is in a dense or sparse area of the network. Furthermore, it may be able to use this information to determine whether it is a leaf node, where a rebroadcast is wasteful, since no new nodes will be covered. Depending on the NWB algorithm being used, the density can be further defined as two different values: (1) The physical density: this consists of the number of neighboring nodes; (2) The NWB density: this is the number of neighboring nodes that perform a NWB transmission. If not all neighboring nodes transmit, it is not useful to track all physical neighbors.

To explain how SAR works in more detail, in the following, we present pseudo-code for SAR integrated with the SBA algorithm. The reader may wish to refer to the SBA implementation in Section 4 to compare against the version with SAR.

```

Let T1 be the SBA Timer
Let T2 be the SAR timer
Let L be the list of nodes that sent a particular broadcast
Let Nx be the neighbor list for node X
Let U be the list of uncovered nodes

```

When a source sends a packet, it is always broadcast.

When a node A receives a broadcast from node S:

```

If ({Na} > {Ns}) Then
  If T1<P> Then // If a timer exists for this packet
    T1<P>.add(P.source); // Add the originator of this broadcast
  Else If T2<P> Then
    T2<P>.add(P.source);
  Else
    T1 := new Timer(P, {Na}-{Ns});
    T1.start();
End If;

```

```

End If;

T1.expire() {
  If ({U} > {N{L}}) Then
    Rebroadcast(P);
    T2 := new Timer(P, {U} - {N{L}});
    T2.start();
  End If;
}

T2.expire() {
  If ({U} - {N{L}}) Then
    If Solution==Probabilistic
      R := Random(0, 1);
      If R > PROBABILITY_THRESHOLD Then
        Rebroadcast(P);
      End If;
    Else If Solution==Counter
      If size(N) < COUNTER_THRESHOLD Then
        Rebroadcast(P);
      End If;
    Else If Solution==Adaptive
      If size(N) == 0 Then
        W := W + 1;
      Else
        W := 0;
      End If;
      If W < INEFFECTIVE_THRESHOLD
        If size(N) != size(BRGs)
          Rebroadcast(P);
        End If;
      End If;
    End If Solution==MacBased
    metric := 1 - (size(N)/size(BRGs));
    metric := metric + macUtilization;
    If metric < MAC_THRESHOLD
      Rebroadcast(P);
    End If;
  End If;
End If;
}

```

Using a purely probabilistic metric to use SAR, coverage is slightly increased, while overhead is increased by the percentage used as the probability value P . This approach is a *blind* method of controlling robustness. A node rebroadcasts solely based on a probability, and takes no external observations into account. As a result, nodes perform the SAR broadcast when not necessary, and a critical rebroadcast may not occur.

Using the counter-based metric for SAR, coverage values are increased more than the probabilistic-based approaches, and the overhead fluctuates, based on network conditions. This approach does make use of observed network behavior, and the results are better, with higher coverages and lower overhead. A counter-based approach is *not blind*, but it also *not adaptive*. For example, nodes near the edge of the network (leaf nodes) may not have their counter value met, resulting in them performing the additional SAR broadcast. This broadcast is wasteful, since no new nodes will be covered.

The effect of the problem of leaf nodes broadcasting unnecessarily can be seen at a 0 loss rate in the counter-based

Figure 20: SAR Approaches Comparison, Coverage

Figure 21: SAR Approaches Comparison, Overhead

Figure 22: Coverage - Additional SAR Metrics

Figure 23: Overhead - Additional SAR Metrics

solution. In this case, an oracle implementation of SAR would not increase the redundancy because no losses occur. However, in the simple counter implementation, there is an increase in overhead with SAR even at 0 loss rate.

To address this problem in an algorithm such as AHBP, a node can use its neighbor knowledge to see how strongly connected it is. A node that is a leaf node will not have a high neighbor count. This knowledge can be used to not cause additional SAR broadcasts if the node has heard a number of broadcasts equal to its CDS neighbor count. Furthermore, for static algorithms, such as AHBP, the neighbors which are to rebroadcast are known at the upstream sending node. The sending node can ensure it has heard a transmission from each rebroadcasting CDS node, and cancel the additional SAR broadcast if the broadcast was propagated successfully.

An additional metric to be used is that of the MAC utilization. A node in a section of the network with more traffic will have a higher MAC utilization value, and the chances of a successful broadcast are smaller.

Figures 22 and 23 compare the base AHBP protocol, the counter-based AHBP-SAR protocol, AHBP-SAR with leaf node pruning, and AHBP-SAR with MAC utilization. It can be seen that the leaf node pruning results in coverage at the same level as the counter-based solution, with a lower overhead. This is due to the suppression of broadcasts from leaf nodes that do not cover additional areas of the network. The MAC utilization version adapts to the increasing loss level in the network. Applied properly, this metric can also be valuable. Since each metric is valuable on its own, they are included in the criteria used for the adaptive SAR protocols.

The *adaptive* metrics cover cases where a counter-based metric does not perform well. They adjust according to more external network indicators, such as the observed network traffic load. Figures 20 and 21 show the coverage and overhead for the three approaches in a sparse 30 node environment, with modifications to the AHBP protocol. In these graphs, it can be seen that the probabilistic approach has the poorest coverage, along with an overhead that does not adapt to the network conditions. The counter-based approach has coverage higher than the probabilistic approach, along with an overhead that tends to increase as the loss rate increases in the network. Lastly, the adaptive approach has the best overall coverage, and an overhead that is increased as the loss rate increases in the network. This overhead starts out around the level of the base AHBP protocol, and stays below the other approaches until the loss rate is

sufficiently high. Even at very high loss rates, the overhead is below the level of flooding, which is constant at a value of 1.

Figure 24: Coverage with SAR, 30 Nodes

Figure 25: Overhead with SAR, 30 Nodes

Some sample results obtained with SAR can be seen in Figures 24 and 25. The overall coverage is increased greatly, with a small increase in overhead. Some broadcasts are critical, and the extra overhead incurred in order to ensure the transmission is successful is worthwhile. For instance, in a network that is subdivided into two distinct sections which are only connected by a single link, it is extremely important the NWB transmissions along the connecting link are successful. Furthermore, for NWB operations where the result is used many times, such as in routing, the extra overhead to discover an efficient route is rewarded every time a packet uses this superior route.

7.2 Directed Broadcast

Directed Broadcast (DB), introduces a new MAC level primitive which signals neighboring nodes that a packet being unicasted is really a public packet [38]. This allows nodes in a CDS tree to use reliable unicast transmissions to propagate NWB information. By keeping the unicast transmissions solely between the nodes in the CDS tree, the overall overhead is kept to a minimum. Furthermore, all nodes in the network are able to receive the packets.

Figure 26: Coverage (Directed Broadcast)

Figure 27: Overhead (Directed Broadcast)

As illustrated in Figure 26, DB provides substantial improvement in coverage, compared to the base protocols. AHBP with DB provides coverage near flooding, and SBA with DB provides coverage above flooding. The overhead of DB (Figure 27) is significantly lower than that of SAR. SBA with DB has overhead only slightly higher than the original protocol at lower loss levels. As losses increase, the overhead of SBA with DB is lower, due to the fact that less packets have been lost, and the protocol doesn't need to try to dynamically recover from those losses. The

overhead of AHBP with DB remains well below flooding, except at very high loss levels, where it costs only slightly more than flooding.

8 Conclusions

Network-Wide Broadcasts (NWBs) are important operations in Ad hoc networks that are used in several routing and group communication algorithms. Existing research has targeted efficient NWB to reduce the amount of redundancy inherent in flooding (the simplest NWB approach). As a result, the NWB becomes more susceptible to loss of coverage due to transmission losses that result from heavy interference or transmission errors. This problem arises because NWBs rely on an unreliable MAC level broadcast operation to reach multiple nodes with one transmission for a more efficient coverage of the nodes. In the presence of interference or transmission errors, this results in nodes not receiving the NWB.

We outline the NWB solution space and explain the properties of the different solutions in terms of overhead, resilience to mobility as well as reliability. We demonstrate the problem using interference from CBR connections as well as using probabilistic dropping of packets (which allows us to control loss rate systematically). As the loss rate rises, and as the density of the network goes down, the coverage achieved by an NWB operation drops. We showed that this is especially true for other NWB algorithms that also rely on MAC level broadcasts because they target reducing the redundancy present in floods, making them more susceptible to losses. Furthermore, we showed that approaches that statically determine the set of forwarding nodes perform much worse than ones that dynamically evaluate whether their rebroadcast is likely to be redundant. As a result, the recent DCB algorithm, while it improves coverage relative to other static approaches, remains substantially more vulnerable to losses than dynamic approaches.

9 Exercises

1. One criticism of robustness control is that the tradeoff is simply increasing the number of retransmissions results in a higher chance of covering the network. Thus, all redundancy control schemes do is increase this redundancy to increase coverage. Discuss whether or not this argument is correct (Hint: is all increase in redundancy the same?)
2. Geocast is the problem of reaching nodes in a prespecified geographical area (assume that nodes know their own location).
 - (a) Suggest an application that would need Geocast
 - (b) Suggest two solutions to implement Geocast (hint, try to adapt solutions for NWB to Geocast).
 - (c) Comment on the robustness of the two solutions and suggest a different improvement for each to address robustness.
3. Repeat question 2 for Multicast (reaching nodes that are members of a multicast group). Only suggest one solution and one improvement
4. Repeat question 2 for Anycast (reaching any one node among the members of an anycast group).
5. Of the NWB protocols covered, which are most resilient to mobility? Explain. What can be done to improve operation in the presence of mobility?
6. None of the protocols discussed considered the use of forward error correction; encoding sufficient redundancy per packet or across packet to enable recovering from errors. Discuss the merits/demerits of such an approach. What type of application would it work best for?

References

- [1] T. Clausen and P. Jacquet, "Optimized link state routing protocol," Internet Draft, Internet Engineering Task Force, Oct. 2003, <http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-11.txt>.
- [2] S. Das, C. Perkins, and E. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proc. of INFOCOM 2000*, Mar. 2000.
- [3] M. Lewis, F. Templin, B. Bellur, and R. Ogier, "Topology broadcast based on reverse-path forwarding (tbrpf)," Internet Draft, Internet Engineering Task Force, June 2002, <http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-06.txt>.
- [4] E. Royer and C. Perkins, "Multicast operation of the ad hoc on-demand distance vector routing protocol," in *Proc. of the ACM International Conference on Mobile Computing and Networking (MobiCom'99)*, 1999, pp. 207–218.
- [5] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of ACM/IEEE International Conference of Mobile Computing and Networking (MOBICOM'99)*, Sept. 1999.
- [6] K. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 2002)*, 2002, pp. 157–164.
- [7] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 2003)*, 2003, pp. 222–232.
- [8] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. of SIGCOMM 2004*, Sept. 2004.
- [9] D. De Couto, D. Aguayo, B. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," in *Proceedings of the First Workshop on Hot Topics in Networks (HotNetsI)*, Oct. 2002.
- [10] T. Rappaport, *Wireless Communication: Principles and Practice, 2nd Edition*, Prentice-Hall, 2001.
- [11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless lan's," in *Proc. SIGCOMM '94*, 1994, pp. 212–225.
- [12] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to hidden terminal problems in wireless networks," in *Proceedings of SIGCOMM 1997*, 1997, pp. 39–49.
- [13] B. Crow, I. Widjaja, J. Kim, and P. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, pp. 116–126, Sept. 1997.
- [14] T. S. Rappaport, *Wireless Communications, Principles, and Practice*, Prentice Hall, 1996.
- [15] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," *IEEE Transactions on Computers*, vol. 52, no. 5, pp. 545–557, 2003.
- [16] D. Scott and A. Yasinsac, "Dynamic probabilistic retransmission in ad hoc networks," in *Proc. of the International Conference on Wireless Networks (ICWN'04)*, June 2004.
- [17] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. of INFOCOM 2002*, June 2002.
- [18] B.N. Clark, C.J. Colbourn, and D.S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.

- [19] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad-hoc networks using a spine," in *Proc. International Conference on Computers and Communications Networks*, Sept. 1997.
- [20] C.-R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, Sept. 1997.
- [21] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," in *Proc. IEEE Hawaii International Conference on System Sciences (HICCS)*, Jan. 2001.
- [22] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, 1999, pp. 7–14.
- [23] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless adhoc networks," in *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.
- [24] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proc. of MobiHoc 2000*, 2000.
- [25] W. Peng and X. Lu, "AHBP: An efficient broadcast protocol for mobile ad hoc networks," *Journal of Science and Technology (Beijing, China)*, 2002.
- [26] W. Lou and J. Wu, "Double-covered broadcast (dcb): A simple reliable broadcast algorithm in manets," in *Proc. of INFOCOM 2004*, 2004.
- [27] UC Berkeley/LNBL/ISI, "The ns-2 network simulator with the cmu mobility extensions," 2002, <http://www.isi.edu/nsnam/ns/>.
- [28] Mesh Networks Inc., "Mesh networks technology," 2004.
- [29] C. Chiang, *Wireless Network Multicasting*, Ph.D. thesis, University of California, Los Angeles, 1998.
- [30] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proc. of INFOCOM 2003*, 2003, (forthcoming). Available from <http://www.eecs.umich.edu/~mingyan/pub/random-waypoint.pdf>.
- [31] C. Bettstetter and C. Wagner, "The spatial node distribution of the random waypoint mobility model," *Mobile Ad-Hoc Netzwerke, 1. deutscher Workshop uber Mobile Ad-Hoc Netzwerke (WMAN 2002)*, pp. 41–58, 2002.
- [32] MIT, "MIT Roofnet," 2002.
- [33] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," in *Proceedings of IEEE MILCOM 2001*, Oct 2001.
- [34] K. Viswanath and K. Obraczka, "An Adaptive Approach to Group Communications in Multi-Hop Ad Hoc Networks," in *Proc. of International Conference on Networking*, 2002.
- [35] Mashud Kabir, "Region-Based Adaptation of Diffusion Protocols in MANETs," M.S. thesis, University of Stuttgart, Germany, Nov. 2003.
- [36] Stefan Pleisch, Mahesh Balakrishnan, Ken Birman, and Robbert van Renesse, "Mistral:: efficient flooding in mobile ad-hoc networks," in *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, New York, NY, USA, 2006, pp. 1–12, ACM Press.
- [37] P. Rogers and N. Abu-Ghazaleh, "Towards Reliable Network Broadcast in Mobile Ad-Hoc Networks," Tech. Rep., SUNY – Binghamton, Feb. 2004.

- [38] P. Rogers and N. Abu-Ghazaleh, "Directed Broadcast: A MAC level Primitive for Robust Network Broadcast," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2005)*, Aug. 2005.