

A Decentralized Approach to Minimum-Energy Broadcasting in Static Ad Hoc Networks

Chris Miller and Christian Poellabauer
Department of Computer Science and Engineering
University of Notre Dame
miller.444@nd.edu, cpoellab@cse.nd.edu

Abstract—Broadcasting is a commonly used feature in wireless networking, e.g. for file distribution, re-tasking, event notification, or miscellaneous maintenance. Due to the limited resources of most wireless ad hoc and sensor networks, minimizing the cost of broadcasts is of utmost importance. This has led to work in the minimum energy broadcasting problem. Solutions to this issue have primarily focused on topology control algorithms or broadcast oriented protocols. Many of these solutions are based on a centralized protocol, where a node has global topology knowledge of the network. However, this information is typically not available in ad hoc applications. Decentralized approaches to this problem have not been able to match the energy efficiency of centralized methods. Both centralized and decentralized approaches have relied upon locality information to estimate link cost. This method is unreliable and can result in poor network models. In this paper, we will describe a new distributed approach to the minimum energy broadcasting problem. Our protocol constructs a broadcast tree in a distributed fashion using only local information. It uses link quality measurements to more accurately estimate link cost, which makes it adaptable to any environment. We show by simulation that our protocol is capable of constructing a tree that is near to centralized approaches in energy cost, without requiring an excessive overhead cost.

I. INTRODUCTION

Wireless sensor networks and ad hoc networks are a collection of small wireless devices, typically battery powered, that may be static or mobile, and may configure in an ad hoc fashion. Broadcasting is a commonly used feature in wireless networking. It may be used for file distribution, re-tasking, event notification or miscellaneous maintenance. The simplest method of accomplishing a broadcast is to flood the message. That is, each node that receives a message for the first time will re-broadcast it to all of its neighbors. This can be an extremely costly operation in terms of network energy usage. It can also result in a high percentage of collisions which could prevent full dissemination. Due to the limited resources of most wireless devices, minimizing the cost of broadcasts is of utmost importance to extending network lifetime.

There have been numerous approaches to reducing the energy cost of broadcasting in wireless networks. The omnidirectional nature of wireless radios provide opportunities for reducing transmission cost, but also make it more complicated to provide optimal solutions. One common approach to reducing transmission cost for broadcasts is to limit the number of forwarding nodes. As opposed to flooding, where all nodes

would forward the packet, a limited subset of nodes would be selected which provide full coverage of the network. These methods reduce the total energy expended in a broadcast by reducing the total number of transmissions. This also reduces contention and collisions, which can make broadcasts more reliable and reduce recovery messages. The set of forwarding nodes may be determined using probabilistic methods, which require little to no information about the topology, or with the use of minimum spanning set algorithms, which guarantee coverage of all nodes. These methods, however, assume that all nodes will be broadcasting at their fullest power. Due to the propagation loss of radio signals, it is actually less energy expensive to make multiple short range transmissions rather than one long range transmission to cover the same length. Therefore, minimizing total transmission cost would provide better results than minimizing transmissions. This also distributes the energy cost of a broadcast over a larger set of nodes, which can prevent the disconnection of a network due to power drain among a subset of nodes.

The minimum energy needed to broadcast a packet to all nodes in a network from a designated source can be found by adjusting the transmission power of each node's radio to achieve a fully connected tree with the minimum transmission cost. This is known as the minimum-energy broadcast tree problem, and it has been shown to be NP-complete [1, 2]. Several sub-optimal approximations have been proposed based upon minimum spanning trees [2, 3, 4], but these have required global knowledge of the topology. In many typical wireless networks, however, there would not be a global knowledge of the topology available at any centralized node. Distributed and localized approaches have been made [5, 6], but these have not been able to match the performance of the centralized approaches. Furthermore, both centralized and de-centralized approaches have used location information for determining the transmission power needed to maintain links. This is not a reliable method of determining link cost, as shown by [7, 8]. In this paper, we propose Dynamic Broadcast Incremental Power (DynaBIP), which constructs a broadcast tree in a distributed fashion using only locally available information and that contained within packet headers. No locality information is needed, as DynaBIP uses received signal strength measurements for the estimation of all link costs. Our goals

in the design of this protocol are to provide a distributed protocol which can construct a broadcast tree which nears the energy efficiency of centralized approaches. The overhead cost of construction should be minimal, so that the amortized energy cost of DynaBIP for session-based broadcasts is still near that of centralized approaches. Simulation will show that DynaBIP performs closely to an ideal centralized approach, with an acceptable level of overhead cost for tree construction.

II. RELATED WORK

One approach to dynamically generating a broadcast tree in a distributed fashion is to use a modified version of topology control algorithms to reduce forwarders and adjust transmit powers. Cartigny et al. proposed RBOP [9] and LBOP [10] which used a neighbor elimination scheme (NES) for a reduced neighbor set. RBOP used the relative neighborhood graph (RNG) [11] to provide the limited set of neighbors that each node would monitor in the NES to determine if it needed to forward the packet. A node which did forward the packet would adjust its transmission power to the minimum needed to cover the uncovered nodes. LBOP used an identical algorithm, but used the local minimum spanning tree (LMST) [12], which provides a smaller subset. TR-LBOP [13] is built upon LBOP, but it aims for an optimal transmit radius in transmit power selection. These broadcast protocols need only local position information which can be provided by immediate neighbors, however they are not able to provide the same level of energy efficiency as centralized protocols, and all NES schemes will result in some redundant transmissions.

Many approaches to the minimum-energy broadcast problem have been based on minimum spanning trees, such as BIP [3], MLE [4], EWMA [2], and MWIA [14]. These are centralized protocols. They require global information of the network topology at a single node in order to perform the algorithm. Most wireless networks, however, do not have this information available. To provide locality information, GPS devices would need to be available to each node, or a localization routine would have to be performed. Each of these options has implications for power consumption, though, and the accuracy of either approach could impact performance.

The most widely referenced solution to the minimum-energy broadcast tree problem is BIP [3]. BIP is a centralized protocol which constructs a broadcast tree using a formula similar to Prim's algorithm for constructing a MST. It adds nodes to a tree rooted at the source one by one, but instead of using link cost it uses incremental cost at each step. This modification allows BIP to benefit from the wireless multicast advantage. That is, if a link from node i to node k has cost P_{ik} , and node i is already transmitting at a power great enough to reach node j , then node i would only have to increase its transmit power by the incremental cost $P_{ik} - P_{ij}$ to reach node k . At each step of the algorithm, the link which provides the lowest incremental cost is added to the tree. Upon completion of the algorithm, a broadcast tree rooted at the source which includes all nodes

in the network will be constructed along with the necessary transmission power of all non-leaf nodes. Wieselthier et al. also provide a sweep algorithm that can be performed as needed following the initial tree construction to remove any redundant transmissions and reduce transmission power where possible. Practice has shown that two sweeps is typically sufficient to obtain best results.

Distributed and localized approaches to BIP have been made. Wieselthier et al. proposed two distributed versions, Dist-BIP-A and Dist-BIP-G [5, 15]. In Dist-BIP-A, a source node constructs a BIP tree of its immediate neighbors using locality information of its local neighborhood. It then broadcast its local tree at the power sufficient to reach all of its neighbors. Each node that receives the broadcast will then construct its own local tree, incorporating the tree forwarded in the packet, and broadcast it to its neighbors. Once all nodes have forwarded their local copy of the tree, the global tree will have been constructed. Many redundant transmissions will have been made in the construction phase, though, which could lead to a high number of collisions in dense networks. The transmit power used to forward the tree is also inefficiently selected, and could lead to further contention. Also, the order in which nodes add links to the tree is somewhat arbitrary, as nodes select the first copy they receive as the parent. Dist-BIP-G sought to reduce some of the contention by having nodes designate a set of gateway nodes. This set of gateways is the set of neighbors that are selected as forwarding nodes by a local BIP tree construction in order to reach all 2-hop neighbors. Only the nodes designated as gateways will broadcast a local BIP tree, which will reduce the number of forwarded packets, while maintaining coverage. This requires 2-hop locality information rather than just the immediate neighborhood, so some additional overhead is incurred. The tree constructed from these distributed protocols is not able to match the energy efficiency of the centralized protocol, and the overhead cost of the construction is high. Also, there are no recovery mechanisms in the tree construction process to ensure that no nodes are left out of the broadcast tree.

LBIP [6] is a localized approach to BIP. It uses a similar approach as Dist-BIP, using 2-hop locality information which it has gathered from neighbors. The source node constructs a local BIP tree for its 2-hop neighborhood and includes the local tree with the broadcast of the packet. Nodes which are designated as senders in the tree will construct their own local tree building upon the tree provided. Each node transmits at the power designated by their local BIP tree calculation. This method allows the broadcast tree to be constructed dynamically for each broadcast, without the setup cost. The energy efficiency of the tree is not as good as that which can be accomplished with a distributed protocol, however. Also, conflicts in the designation of sending nodes could lead to an incomplete broadcast tree. Therefore, a NES scheme has to be included to ensure that all nodes receive the broadcast. This adds significant redundant transmissions, which hinders

LBIP's ability to approximate BIP.

Each of these protocols relies on the ability to calculate the transmit power necessary for a link based on distance. Providing the framework necessary to supply locality information for distance calculation can impose a significant amount of overhead on a network. Assuming that accurate locality information is available, the use of distance measurements to calculate necessary transmit power is unreliable. This is because the path loss properties of each environment are different, so there is no standard calculation that can be performed without modeling the environment of the implementation. Also, the path loss properties can vary among links within the same network, due to variations in obstructions, orientations, etc. The distributed protocols have difficulty achieving the same level of efficiency as the centralized protocols. One reason for this is that the decision process for adding links to the tree is placed on the parent nodes. However, the parent nodes have a limited view of the paths available to each of their neighbors, so they are not able to make the optimal choice for selecting children. Our contribution is to propose a new distributed protocol, DynaBIP, which uses signal strength measurements to determine link costs, and places the decision process during tree construction on the child nodes. This makes DynaBIP adaptable to any environment, and tree construction is based on a more accurate model of the network, allowing DynaBIP to achieve the full potential of its energy savings while maintaining connectivity.

III. DYNAMIC BROADCAST INCREMENTAL POWER PROTOCOL

BIP and related protocols provide a good approximation of the minimum-energy broadcast tree. One related requirement, however, is that they all use location information for determining link cost. Location information may be obtained with a GPS device, or by using a localization algorithm, but these can be expensive in energy cost. In addition, there may be additional energy costs associated with propagating the locality information to neighboring nodes, or a centralized node. Of greater concern, though, is the unreliability of computing link cost based on distance. It has been shown that link quality is not consistent with distance [7, 8], and that signal propagation can vary widely in different environments. Estimating the pathloss exponent that should be used to model the environment is difficult, since this property differs among implementations and can vary among links within the same network. If the degree of pathloss is underestimated, then the calculated transmit powers will be insufficient, and the broadcast tree will become disconnected and incomplete. If the degree of pathloss is overestimated, the calculated transmit powers will be greater than necessary, resulting in reduced energy efficiency and possibly contention.

Distributed implementations of BIP have difficulty approximating the centralized broadcast tree because the nodes constructing local broadcast trees have limited knowledge of alternate paths to their neighbors. This can result in redundant

or inefficient selections, since a node which may designate itself as a parent is out of immediate range of another potential parent of the child node, and it is therefore unaware of its actions.

We propose a new distributed approach to this problem to address these issues. Dynamic Broadcast Incremental Power (DynaBIP) performs a distributed construction of a broadcast tree using received signal strength measurements to estimate link costs. This allows each node to adapt individually to the signal propagation properties of each link. The link cost estimations are thereby more accurate than can be achieved by path loss calculations, and adaptable to changes. Construction of the tree is managed by the child nodes, instead of the parent nodes, to provide a more complete vision of the available routes. Since child nodes are aware of all potential routes to themselves, they can determine the optimal path better than a parent node working with partial information.

DynaBIP constructs a broadcast tree in a similar manner to BIP. It selects the link which provides the minimum incremental cost at each iteration. For each step, the node and link which provide the smallest incremental cost among the remaining nodes is added to the tree. Without global information, there is no way for nodes to determine which node should be selected for the current iteration. With the use of carefully selected delays based upon overheard signals, though, nodes may be able to approximate the order in which they should be included in the tree. DynaBIP will use the observed cost of each link to determine what order it should add nodes to the broadcast tree. The remainder of this section will describe the construction algorithm for DynaBIP. Following is a list of terms that describe values or data structures needed by each node. RSS refers to the received signal strength, or the power measured at the receiving node's radio upon reception. P_i is the initial transmit power used by node i in the construction phase. This may be assumed to be the maximum transmit power, but alternative options will be discussed later in this section. C_{ij} is the cost of link $i \rightarrow j$, or the transmit power needed by i to reach j . T_i is the transmit power for node i as determined by the DynaBIP algorithm. R_{Thresh} is the receiving threshold necessary to ensure reception. The *Link_Cost* table is a table of all C_{ij} values, stored on node j and indexed by i . The *Transmit_Power* table is a table of T_i values, indexed by i . Each node will also maintain its parent node, a list of its children, and a *Route_Cache* that is indexed by the last hop in the route.

A. DynaBIP tree construction algorithm

The tree construction of DynaBIP can be completed with a single sweeping flood across the network. The source node will initiate the tree construction by broadcasting the first packet. It will include in the header of the packet its transmit power, P_{src} and the route to this node (a null array for the source). Each node that receives this broadcast will compare the RSS to the transmit power, P_{src} , to determine the path loss for this

link. The path loss for a link $i \rightarrow j$ may be depicted as

$$PathLoss_{ij} = P_i - RSS \quad (1)$$

Adding this value to the radio receiving threshold will provide the transmit power necessary to maintain this link. The link cost can then be represented as

$$C_{ij} = R_{Thresh} + PathLoss_{ij} \quad (2)$$

C_{ij} will be saved in the *Link_Cost* table. Each node will store the route in the header in the *Route_Cache*, indexed by the sending node's ID, in this case the source's ID. It will then schedule a rebroadcast of this packet at a delay proportional to C_{ij} in milliWatts. If it receives another broadcast of this packet from a closer node (a lower cost link), it will cancel the previously scheduled rebroadcast, and schedule a new rebroadcast with a delay proportional to the new link cost. The formula for the delay is

$$ForwardingDelay = \beta \times C_{ij}[mW] \quad (3)$$

where β is the delay multiplier. There is a tradeoff between contention and latency of tree construction for the multiplier. In our trials, we found that β values between the range 10 to 100 *ms/mW* provided a good balance between contention and latency. If a collision does occur due to closely scheduled rebroadcasts, a resend will be scheduled at a random backoff. This process is described further in Section III-D.

The node which has the lowest cost link from the source will rebroadcast first. It will include in its header its transmit power, P_i , the route to this node, which will include only the source node, and the transmit power needed by the source to maintain this link, C_{si} . When the source node overhears this broadcast, it will recognize that it has been selected as the parent by the previous node, and will update T_{src} to the link cost provided in the header, C_{si} . It will also add the previous hop to its list of children, and send an acknowledgment to the new child node. Other nodes which overhear the broadcast will store the link cost to the new node, C_{ij} , in their *Link_Cost* table, and cache the new route. They will also update T_{src} in the *Transmit_Power* table to the link cost value in the header, C_{si} . If C_{ij} has a lower cost than the previous link to the source, or if this is the first copy of the packet it has received, it will schedule a rebroadcast at a delay proportional to the cost of the new link. This process is then repeated by each node that receives a copy of the broadcast.

When a node reaches its scheduled rebroadcast time, it will perform a calculation to determine which link among the nodes currently in the broadcast tree will provide the minimum incremental cost. It does this by scanning through the routing cache. For each entry in the routing cache, it compares the current transmit power of the last hop to the link cost from the last hop in the route to itself. The minimum difference among these entries is the route that will be selected as the link for this node's addition to the broadcast tree. The algorithm for selecting the new link to add at node j is shown in Algorithm

Algorithm 1 Minimum Incremental Cost Link Selection

```

minCost ← ∞
for all  $r_i \in RouteCache$  do
  incrementalCost ←  $C_{ij} - T_i$ 
  if incrementalCost < minCost then
    minCost ← incrementalCost
    parent ←  $i$ 
    minRoute ←  $r_i$ 
  end if
end for

```

1, where r_i is an entry in the routing cache with last hop node i . The route which provides this minimum incremental cost will then be selected by the node for the next addition. It will set its parent to the last hop node in this route, and will include this route in the packet header, along with the link cost needed by its parent, C_{ij} . Once it rebroadcasts the packet, it will have added itself and the link to its parent to the distribution tree.

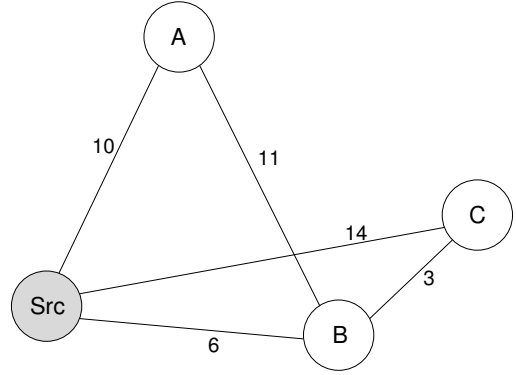


Fig. 1: Example of node selection order of DynaBIP.

The proportional delay is what allows DynaBIP to emulate a global incremental power algorithm in the tree construction process. In each case, the link with the minimum additional energy cost will be the node which is added to the tree next. This concept will continue to work over links of varying hop counts. Consider the topology in Figure 1. The source initiates a broadcast at time unit 0. Node B has the lowest link cost, and will schedule a rebroadcast at time unit 6, representing an incremental cost of 6. Node A will schedule its broadcast at time unit 10, and node C at time unit 14. When node B rebroadcasts at time unit 6, node C will cancel its scheduled broadcast at time unit 14, and will reschedule it for time unit 9, since it has a link cost of 3 from node B. Node A will not alter its scheduled rebroadcast time since the link to B is not less than the link to the source. Node C will then be the second node to rebroadcast, with an incremental link cost of 3. Finally, node A will rebroadcast at time unit 10, with an incremental link cost of $C_{SrcA} - C_{SrcB} = 4$. This is the same ordering of adding nodes and links to the broadcast tree as would be observed by the centralized BIP algorithm. The ordering in

which nodes are added to the broadcast tree globally may not be precisely by incremental link cost, but locally it is a close approximation.

B. Flooding power selection

As mentioned in the previous section, the construction of the DynaBIP tree can be completed in a single flood. The cost of tree construction will then be dependent upon the initial transmit power used by the nodes during the flooding phase. Three variations of the flooding phase were explored in the tree construction process. The significant properties of each method are energy cost, quality of broadcast tree constructed, and completeness of broadcast tree (whether all nodes are included in the tree).

1) *Standard flood*: The simplest method of implementing the flooding phase of the tree construction is to have each node broadcast at full power. This incurs the highest cost of construction, but requires no information of local topology. As long as the network is fully connected, the broadcast tree will be complete. This method will provide the closest approximation of the BIP broadcast tree for low to medium density networks, since all links will be explored. For high density networks, however, flooding at maximum power can lead to excessive contention and collisions, which could hamper the quality of the broadcast tree and lead to additional construction costs for resends. A lower power may be used for the standard flood method to reduce construction cost and collisions, but without a more intelligent algorithm, full connectivity can not be guaranteed.

2) *Optimal transmit radius*: An alternate method was explored to attempt to optimize the transmit power per node while maintaining broad enough coverage to match the quality of the broadcast tree constructed by a standard flood. The optimal initial transmit power was approximated as the power needed to maximize the nodes reached per power unit. This required only single hop local neighborhood information to find the transmit power which maximized this metric. The construction cost were greatly reduced from standard flooding, as well as collisions, but they were still high relative to the energy cost of the broadcast tree. The quality of the tree constructed was nearly identical to the flooding method when it was complete, however it did not guarantee full connectivity.

3) *RNG flood*: The relative neighborhood graph (RNG) [11] provides a fully connected graph with a slightly higher level of redundancy than a MST. By setting the transmit power of each node in the flooding phase to the minimum power needed to cover their RNG neighbors, a near minimal power is used to maintain full connectivity.

The determination of the RNG neighborhood is performed locally. This requires 2-hop neighborhood information at each node. This information may be maintained by including information about each nodes immediate neighbors in beacon or "HELLO" messages, similar to methods used by LBIP [6] and Dist-BIP [5]. Instead of including the location of each of

its neighbors, however, nodes will include the link cost to each of its neighbors. As with the tree construction algorithm, our construction of the RNG is based upon energy cost of a link rather than distance.

This method provides the lowest construction cost of the methods explored. The slightly higher level of redundancy provided by RNG versus MST leads to a fairly high quality broadcast tree. Though not as optimal as a standard flood in a collision-free environment, it performed very well in comparison to a standard flood in an environment with contention and collisions. Due to the quality of the broadcast tree constructed from this method, and the energy savings in the construction phase, the RNG flood was selected as the method of construction. All results presented for DynaBIP will utilize this method.

C. Sweep method

The BIP algorithm includes a method of performing a sweep following the initial construction of the broadcast tree in order to remove redundant links due to the wireless broadcast advantage [3]. The sweep operation may be performed an unlimited number of times, but practice has shown that two iterations is usually sufficient to reach the optimal level. To mimic this operation in a distributed fashion, each node will perform a sweep to determine if an alternate parent may be chosen to reduce transmission cost. After determining a parent and rebroadcasting the initial packet, each node will schedule a sweep operation following a fixed delay period. This delay period will be long enough to allow the initial tree construction process to be completed among its immediate neighborhood. Following this delay, it will perform the same scan of its routing cache as it did during the initial construction phase. If it finds that another potential parent has increased its transmit power to a level high enough to reach itself, it will change its parent to the new candidate and broadcast this selection at a transmit power high enough for the new and prior parent to receive it. Upon receiving this broadcast, the prior parent will remove the sending node from its child list, and adjust its transmit power to the minimum needed to cover its remaining children, if any.

D. Error handling

Ensuring that the broadcast tree constructed includes all nodes in the network is an important qualification for our protocol. In order to accomplish this, acknowledgment messages are used. When a node overhears a message that indicates it has been selected as a parent, it will send an acknowledgment packet to the child node to verify that it has added it to its child list. When a node rebroadcasts the initial packet, it will schedule a resend of the packet at a designated backoff time. During this time interval, it will listen for an acknowledgment from its parent node indicating that it has received the packet and added the node as one of its children. If an Ack from the parent is not received during this time, the packet will be resent. Resends will take place after random delays in order to

avoid synchronization of contending nodes. This will ensure that all links are added as expected in case of a dropped packet or collision. An implicit acknowledgment process is also used to ensure that all nodes are included in the construction phase. Each node monitors its RNG neighbors for broadcasts. If any RNG neighbor does not rebroadcast the message, then it may assume this neighbor did not receive a copy of the message, and will resend the message to them. This will ensure that all nodes are included in the tree construction.

IV. PATH LOSS ESTIMATION

As described in the previous section, DynaBIP adapts to the pathloss properties of each link. This is because the link cost calculation is based upon received signal strength measurements, as opposed to a model of standard path loss. In order to obtain the signal strength measurement, typically the RSSI reading of the radio must be read. Many consider RSSI a poor indicator of link quality, just as distance is a poor indicator of path loss. This perception is largely based on studies which show that RSSI values are not strongly correlated with distance, and may fluctuate over different time frames [7, 8]. This is due to variations in the environmental factors which may change over time. This does not indicate, however, that RSSI values are not consistent among individual links for shorter time frames. It also does not take into consideration that newer radios may provide stronger link quality indicators. In [16], a testbed of Berkely Zigbee nodes based on IEEE 802.15.4 radios was used to evaluate if RSSI is a strong indicator of link quality. They found that the RSSI values had a small variance over time for individual links. This study shows that RSSI may provide a good estimation of the quality of an individual link for short time frames, which is sufficient for session-based broadcasts. Making use of this information on a link-by-link basis allows DynaBIP to optimize the tree construction for all variations within the network.

V. SIMULATION

To evaluate DynaBIP, it was implemented in the Jist/Swans simulation environment [17, 18], a scalable wireless ad hoc network simulator based in Java. Swans provides a full representation of the complete network layer model, with accurate representations of a wireless environment, including path loss, environmental noise and collision interference. Each node in the simulation was implemented with an 802.11 radio, with a maximum transmit power of 11 dBm. The environment was modeled using free space path loss with a pathloss exponent of 2 or 4. A fixed field size was used for each simulation, with an increasing number of nodes to provide an increasing density. Each iteration of simulations is performed 100 times, with a differing random node placement for each run. Only trials with a fully connected network topology were included in the results, which primarily affected the trials with node density 50. The results provided are arranged by average node degree. Due to the random placement of nodes in Jist/Swans,

the degree of each node can differ drastically from the average node degree. Figure 2 depicts the average node degree for each node density, as well as the minimum and maximum node degree.

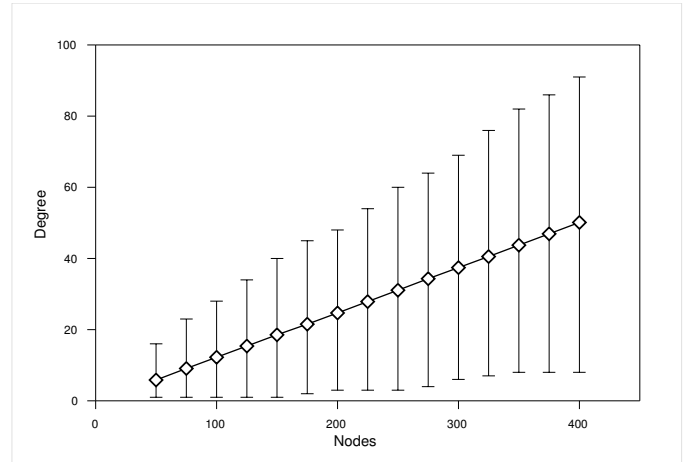


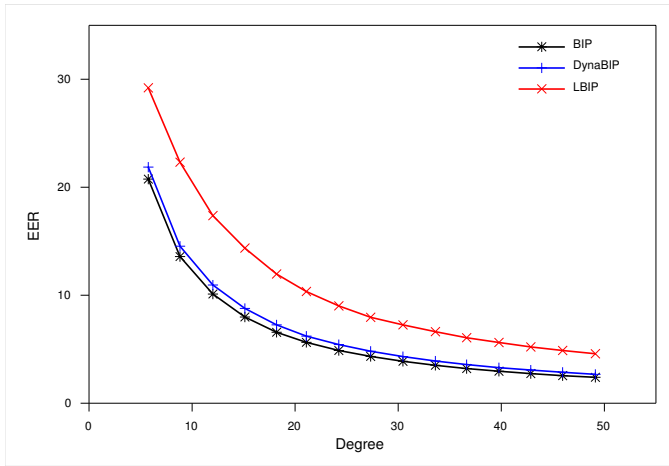
Fig. 2: The average degree of connectivity by node density. The values plotted show the minimum and maximum node degree for each density.

For comparison, BIP [3] and LBIP [6] are also implemented in Jist/Swans. BIP provides an optimal benchmark for a distributed solution. The simulations for BIP do not include actual radio transmissions. It is merely a calculation of the BIP tree based on the same topologies as the DynaBIP simulations, assuming that the centralized node in BIP knows the precise energy cost of each link. This implementation of BIP utilizes two sweep operations to provide the minimum cost BIP tree possible. LBIP is provided to allow a comparison to a strong distributed implementation of BIP. LBIP is a localized protocol, so it does not incur the construction cost of a distributed protocol, but provides a good approximation of the energy cost of BIP. The implementation of LBIP uses the NES scheme for RNG neighbors to ensure a fully connected broadcast tree. The primary metrics used to evaluate the protocols are tree construction cost and energy cost of the broadcast tree. As in [6], the energy cost will be represented as EER, the Expended Energy Ratio. EER is the ratio of energy consumed by a protocol in comparison to the energy that would have been consumed by a blind flood. It is defined as

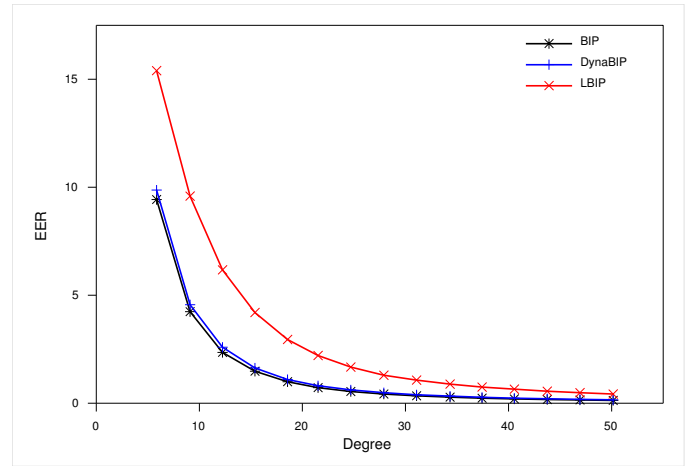
$$EER = \frac{E_{protocol}}{E_{flooding}} \times 100. \quad (4)$$

VI. RESULTS

The first set of graphs in Figure 3 show the total energy cost of the constructed broadcast tree by average node degree. The first graph shows the resulting tree when the environment is modeled with a pathloss exponent of 2. The second graph displays the broadcast tree cost for an environment modeled

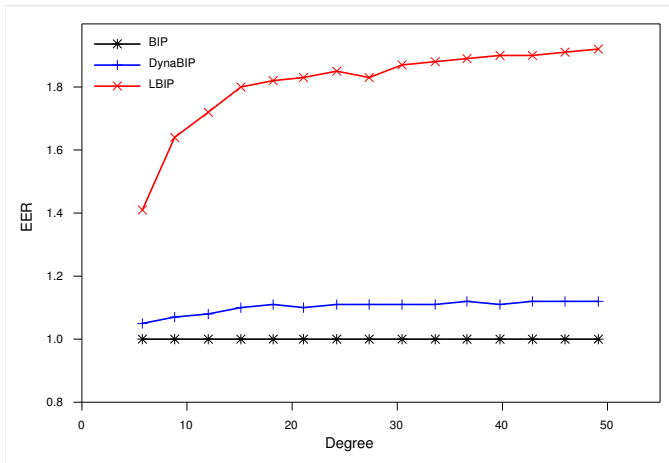


(a) Path loss exponent: 2

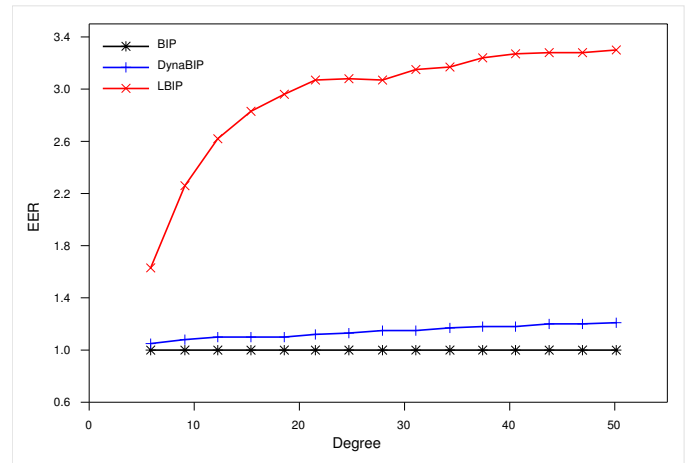


(b) Path loss exponent: 4

Fig. 3: Total broadcast tree energy cost plotted by average node degree.



(a) Path loss exponent: 2



(b) Path loss exponent: 4

Fig. 4: Total broadcast tree energy cost normalized to the energy cost of BIP.

with a pathloss exponent of 4. These results show that DynaBIP performs very closely with the centralized BIP protocol. It is important to note here that the results for BIP and LBIP assume that they are able to perfectly estimate the required transmission power to maintain a link based on locality information. Therefore, the plots for BIP and LBIP may be considered ideal results. DynaBIP uses the actual observed link cost, so it will perform consistently regardless of variations in the pathloss properties of the environment. Despite this, DynaBIP still performs very strongly in comparison to the ideal plots of BIP and LBIP. Figure 4 is another graph of the energy cost of the broadcast tree, normalized to the cost of BIP. It can be more easily seen here how DynaBIP maintains a close approximation of centralized BIP even in very dense networks.

DynaBIP is designed for session-based broadcast, that is, multi-packet broadcasts. There is an overhead cost associated

with the construction phase of DynaBIP. In order for DynaBIP to be viable, the construction cost must be small enough that it can be quickly amortized over a small number of packets in order to provide a total energy cost for a broadcast session which is lower than alternatives. This includes all acknowledgment messages and any recovery messages that are necessary, since reliability of tree construction is considered a priority of our protocol. These control messages are not factored into the cost of BIP and LBIP, since they are not part of the protocols. However, similar control messages would be necessary for BIP and LBIP if reliability of tree construction is a requirement.

BIP does not require any transmissions for tree construction, since it is computed on a central node, however it will be necessary to disseminate the constructed broadcast tree to the nodes in the network, which may include control messages to ensure reliability. This cost has not been factored into any

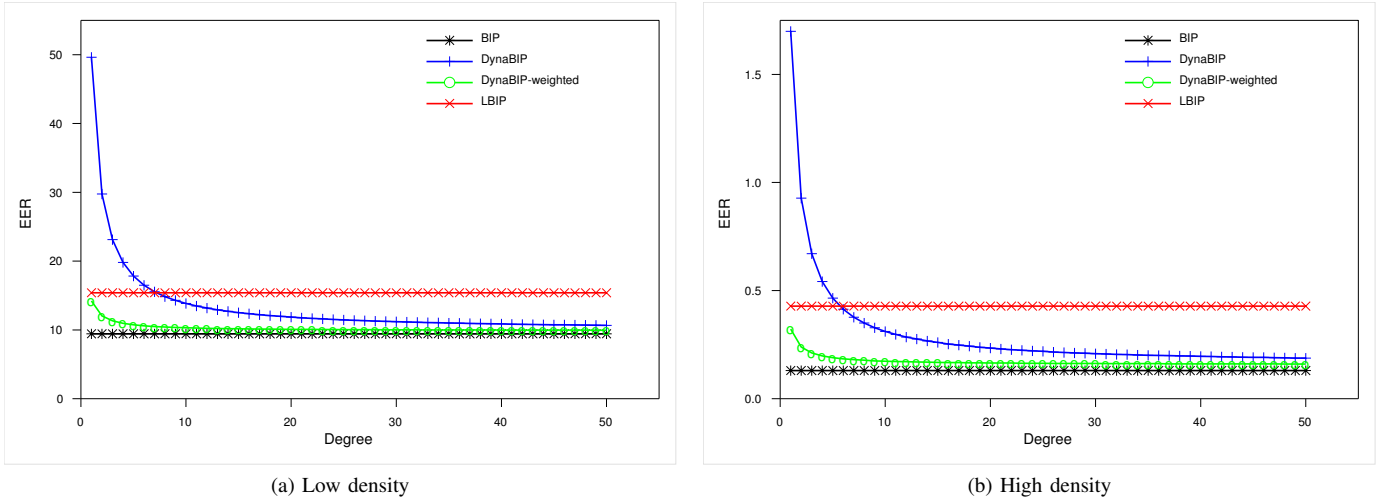


Fig. 5: Total broadcast session energy cost plotted by number of packets. For DynaBIP, this includes all control messages necessary for tree construction. DynaBIP-weighted is based on a chunk size of 1000 bytes.

plot of BIP. LBIP does not have a construction cost because it is a localized protocol which constructs the tree dynamically. Therefore, the broadcast tree cost is the construction cost. As a result, LBIP may provide a more efficient alternative for single packet broadcasts. For session-based broadcast, however, where multiple packets are distributed, the energy savings of a more efficient broadcast tree will provide the best overall savings. Figure 5 shows how the energy cost per packet compares for increasing file sizes. For DynaBIP, this is the energy cost of construction, including any acknowledgment and recovery messages, plus the energy cost for each broadcast. The first graph is for a low density network, average node degree 5.85, with a path loss exponent of 4. The second graph is for a high density network with an average node degree of 50.14. As the number of packets in the broadcast increases, the overhead of construction is amortized over more packets, providing a higher level of efficiency per packet. If the maximum packet size is 1 KB, a simple 50 KB file will require 50 packets in a single session.

There are two plots for DynaBIP in Figure 5. The first plot assumes that all packet sizes will be equal, including those in the construction phase. This is not entirely accurate, however, since construction phase packets need only include the header information, which is small. The construction phase also includes acknowledgment and recovery messages, which are also relatively small. This would reduce the impact of the higher transmission cost of the construction phase since only a fraction of the number of bytes would be transmitted. The second plot, DynaBIP-weighted, shows how the session-based broadcast cost would compare if the packets were weighted by their actual packet size, including any network and MAC layer headers. This plot is based on a chunk size of 1000 bytes. The weighted DynaBIP performs close to the efficiency level of centralized BIP after only a few packets, and provides

lower total energy cost than LBIP after only a single packet. Even under the unweighted DynaBIP plot, the overall energy efficiency of DynaBIP exceeds LBIP after just a handful of packets.

VII. CONCLUSIONS

Through simulation we have shown that DynaBIP is capable of constructing broadcast trees very close in efficiency to an ideal measure of centralized BIP, with an acceptable construction cost that may be amortized over a small number of packets. Our results have shown that DynaBIP provides a closer approximation of BIP for session-based broadcast than LBIP. Most importantly, DynaBIP has shown that it is capable of performing well in comparison to ideal evaluations of centralized BIP and distributed variations, without relying on any information about the environment. BIP and LBIP depend heavily on their ability to accurately estimate the transmit power necessary to maintain links. If these protocols underestimate this cost, the broadcast trees will become fractured and incomplete. If they overestimate the link cost, the protocols will not provide the level of energy efficiency expected. The strength of DynaBIP is its ability to adjust to the pathloss properties of any environment, even when those properties differ among links in the same network. Also, by placing the decision process of tree construction on the child nodes, rather than the parent nodes, DynaBIP is able to construct a more efficient tree than other distributed algorithms. This is because child nodes have a more complete picture of the routes available to them than their potential parents.

While it was not evaluated in this paper, DynaBIP may be easily tailored to handle different cost metrics in the decision process of tree construction. For instance, there may be a fixed cost associated with any node transmitting due to resources needed to power up the radio, perform processor computations, etc. These costs can easily be factored into the decision process

by adding the fixed cost to the estimated link cost. Also, many wireless devices offer only a limited number of discrete power level adjustments. This can be factored into link cost evaluations by providing a ceiling function which rounds the link cost to the next highest power level. For each of these situations, the algorithm will perform the same as before, using the desired adjusted cost metrics. DynaBIP could also be tailored to factor in other desired metrics in the cost decision process, such as remaining battery power. This makes it an easily extendable distributed protocol for achieving minimum-energy cost broadcast trees with multiple decision criteria.

REFERENCES

- [1] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks." in *In Proceedings of 3rd International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM, 2002, pp. 112–122.
- [2] M. Zagalj, J.-P. Hubaux, and C. C. Enz, "Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues." in *In Proceedings of the 8th International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2002, pp. 172–182.
- [3] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks." in *In Proceedings of the 19th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000, pp. 585–594.
- [4] M. X. Cheng, J. Sun, M. Min, and D.-Z. Du, "Energy-efficient broadcast and multicast routing in ad hoc wireless networks," in *In Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, April 2003, pp. 87–94.
- [5] J. Wieselthier, G. Nguyen, and A. Ephremides, "Distributed algorithms for energy-efficient broadcasting in ad hoc networks," in *In Proceedings of the Military Communications Conference (MILCOM)*, vol. 2, Oct. 2002, pp. 820–825 vol.2.
- [6] F. Ingelrest and D. Simplot-Ryl, "Localized broadcast incremental power protocol for wireless ad hoc networks." in *In proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC)*. IEEE Computer Society, 2005, pp. 28–33.
- [7] M. Souryal, L. Klein-Berndt, L. Miller, and N. Moayeri, "Link assessment in an indoor 802.11 network," in *In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, 0-0 2006, pp. 1402–1407.
- [8] S. Z. Erdogan and S. Hussain, "Using received signal strength variation for energy efficient data dissemination in wireless sensor networks." in *In Proceedings of 18th International Conference on Database and Expert Applications (DEXA)*. IEEE Computer Society, 2007, pp. 620–624.
- [9] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized minimum-energy broadcasting in ad-hoc networks." in *In Proceedings of the 22nd Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. IEEE Communications Society, 2003.
- [10] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, "Localized lmsr and rng based minimum-energy broadcast protocols in ad hoc networks." *In Ad Hoc Networks*, vol. 3, no. 1, pp. 1–16, 2005.
- [11] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *In Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.
- [12] N. Li, J. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," *In IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, May 2005.
- [13] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, "Target transmission radius over lmsr for energy-efficient broadcast protocol in ad hoc networks," in *ICC*, vol. 7, June 2004, pp. 4044–4049 Vol.7.
- [14] M. X. Cheng, J. Sun, M. Min, Y. Li, and W. Wu, "Energy-efficient broadcast and multicast routing in multihop ad hoc wireless networks." *In Wireless Communications and Mobile Computing*, vol. 6, pp. 213–223, 2006.
- [15] J. Wieselthier, G. Nguyen, and A. Ephremides, "The energy efficiency of distributed algorithms for broadcasting in ad hoc networks," in *In Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, vol. 2, Oct. 2002, pp. 499–503 vol.2.
- [16] K. Srinivasan and P. Levis, "Rssi is under appreciated," in *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*, May 2006.
- [17] R. Barr and Z. Haas, "Jist/swans website," April 2004. [Online]. Available: <http://jist.ece.cornell.edu/>
- [18] R. Barr, Z. J. Haas, and R. van Renesse, "Jist: an efficient approach to simulation using virtual machines: Research articles," *In Software - Practice and Experience*, vol. 35, no. 6, pp. 539–576, 2005.