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. Due to the limited resources of most wireless ad hoc and sensor networks, minimizing the cost of commonly used broadcasts is of utmost importance. This has led to work in the minimum energy broadcasting problem. Most solutions require global topology knowledge, however, this information is typically not available in ad hoc applications. Decentralized approaches have been unable to match the energy efficiency of centralized methods. Previous approaches have also relied upon locality information to estimate link cost, which is unreliable. In this paper, we will describe a new distributed approach to the minimum energy broadcasting problem which targets multi-packet broadcast sessions. It constructs a broadcast tree in a distributed fashion using link quality measurements to more accurately estimate link cost. We show by simulation, and confirm through experimentation, that our protocol is capable of constructing a tree that is near to centralized approaches in energy cost.

1 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. Naive approaches to broadcasting can be extremely costly 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.

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 [3, 2, 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, with a minimal overhead which can be amortized over a short broadcast session.

1.1 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 due to the hidden terminal problem.

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 which require global information of the network topology. Most wireless networks, however, do not have this information available. The most widely referenced solution to the minimum-energy broadcast tree problem is BIP, which has been shown to approximate the optimal solution within a constant ratio [15]. 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.

Distributed and localized approaches to BIP have been made. Wieselthier et al. proposed two distributed versions, Dist-BIP-A and Dist-BIP-G [5, 16]. In Dist-BIP-A, a source node constructs a BIP tree of its immediate neighbors.

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. Dist-BIP-G uses a similar approach, but designates gateway nodes to reduce the number of transmissions in the tree construction process. 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 may be high.

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. This method allows the broadcast tree to be constructed dynamically for each broadcast, without the setup cost. Simulation results showed LBIP performed close to the efficiency of BIP in dense networks, but the need to incorporate a NES scheme for reliability hindered its ability to match the centralized protocol.

1.2 Contributions

We 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. Simulation will show that DynaBIP performs closely to an ideal centralized approach, and these results are supported through experimentation.

2 Dynamic Broadcast Incremental Power

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. 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 (RSS) measurements to estimate link costs. It has been shown that RSS measurements have small variance over short time frames for individual links, and can provide a good approximation of link quality [8, 17, 18]. 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.

2.1 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. If P_i is the initial transmit power used by node i, the path loss for a link $i \rightarrow j$ may be depicted as

$$PathLoss_{ij} = P_i - RSS \tag{1}$$

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

$$C_{ij} = R_{Thresh} + PathLoss_{ij} \tag{2}$$

The link cost, C_{ij}, will be saved in the *Link_Cost* table of node j. Each node will store the route in the header in their *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

$$ForwardingDelay = \beta \times C_{ij}[mW] \tag{3}$$

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.

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 is has been selected as the parent for node i. It will update its transmit power for the broadcast session, T_{src}, to the link cost provided in the header, C_{si}. It will also add i 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 their $Transmit\_Power$ 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 link that will be selected 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 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}.

Algorithm 1 Minimum Incremental Cost Link Selection

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

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.

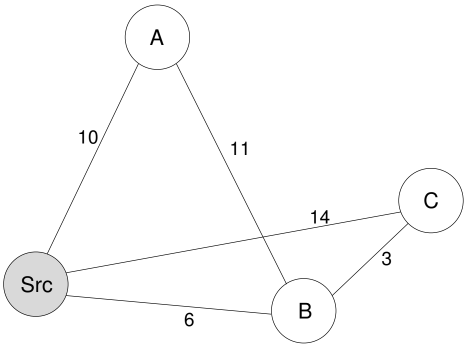


Fig. 1: Example of node selection order of DynaBIP.

2.2 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. 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.

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 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.

The determination of the RNG neighborhood is performed locally. This requires 2-hop neighborhood information at each node. This information may be maintained using beacon 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.

2.3 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 a better parent candidate has increased its transmit power to a level high enough to reach itself, it will select the alternate parent. The previous 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.

2.4 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. If an Ack from the parent is not received during a short time interval, 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.

3 Results

3.1 Simulation

To evaluate DynaBIP, it was implemented in the Jist/Swans simulation environment [19, 20], 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.

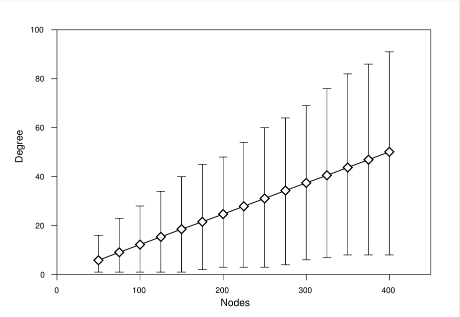
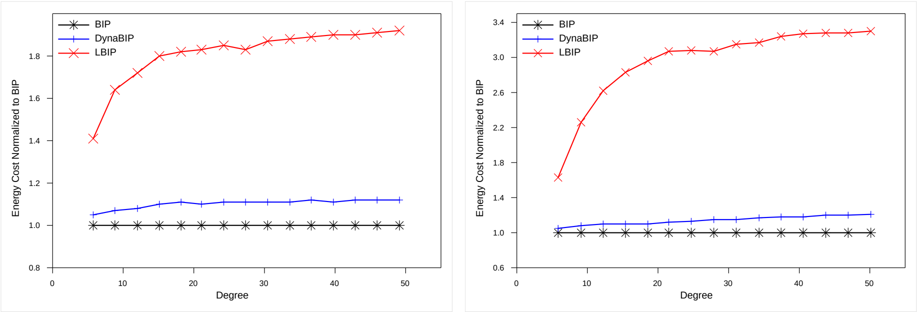


Fig. 2: The average degree of connectivity by node density. The values plotted show the minimum and maximum node degree for each density.

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

$$EER = \frac{E_{protocol}}{E_{flooding}} \times 100. \tag{4}$$

The graphs in Figure 3 show the total energy cost of the constructed broadcast tree by average node degree, normalized to the cost of BIP. The first graph shows the resulting tree when the environment is modeled with a pathloss exponent of 2, and the second 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.

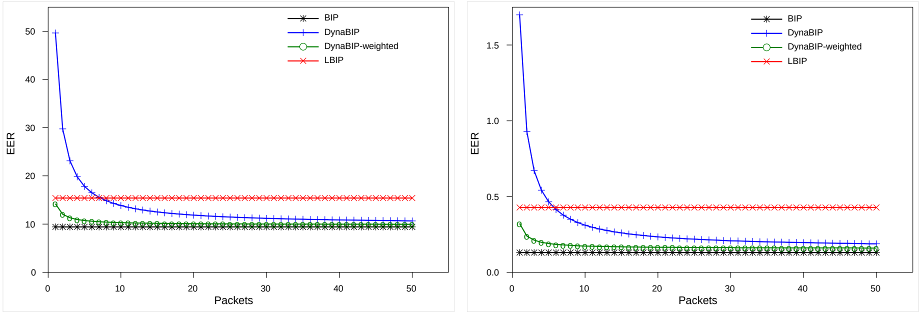


(a) Path loss exponent: 2

(b) Path loss exponent: 4

Fig. 3: Total broadcast tree energy cost normalized to the energy cost of BIP.

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. 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 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 4 shows how the energy cost per packet compares for increasing file sizes. For DynaBIP, this includes the energy cost of construction. 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.



(a) Low density

(b) High density

Fig. 4: 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.

There are two plots for DynaBIP in Figure 4. 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.

3.2 Experimentation

To further validate the results of the simulation, we implemented DynaBIP on a set of Crossbow Stargate devices [21]. The testbed consisted of 12 SPB400CB gateway devices with 802.11b CompactFlash wireless ethernet cards. The wireless cards have a transmit power range of -10 dBM to 13dBm. We divided the transmit power control range into 64 discrete levels, and for each measured the output transmit power in dBm, and the corresponding current in amperes. A plot of these values is shown in Figure 5.

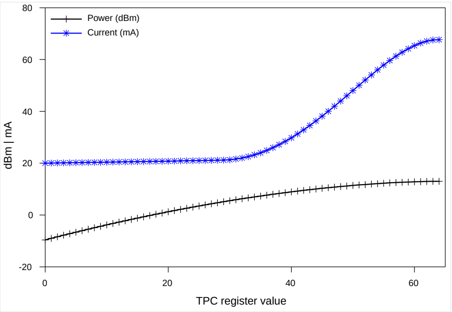


Fig. 5: Plot of radio transmission power in dBm and current in mA for range of values used in transmit power control.

The devices were arranged in a residential indoor environment. For each test, a 26 KB file was distributed. The file was divided into 27 chunks, each 1000 bytes, except the last chunk which included the remainder. In order to ensure strong links which minimize packet errors, the transmit power calculation included a 10 dB buffer to provide a SNR of at least 10 dB. Therefore, the equation for link cost estimation from node i to j in decibels is

$$C_{ij} = TransmitPower_i - RSSI_j + Noise_j + 10dB \tag{5}$$

The file distribution tests were run on the Stargate testbed five times, at various times of alternate days. We logged the calculated link cost for all node pairs for each test. As expected, these values varied among the different tests, in some cases significantly. This is due to changes in the environment during each of the test, such as obstacles which have moved, changes in the wireless environment, etc. Also, distance was not a reliable indicator of path cost, and the path cost of each link is typically not reciprocal. As a result, centralized protocols could not perform as optimally as expected in a real environment, because the path cost estimations used in their calculations are not reliable. This can lead to inefficiency or disconnected components.

For each test, the measured link cost for all node pairs was used to determine the broadcast tree that would have been constructed by a centralized BIP algorithm, if it was provided with the actual link cost calculated in the test. This is provided as a comparison to an ideal scenario, since centralized minimum energy broadcast algorithms such as BIP would typically estimate link cost based on distance. A comparison for each of the trials is provided in Table 1.

Table 1: Total Cost of Broadcast Tree

Trial	BIP	Cascade	%Diff
1	82.96	83.37	0.5%
2	108.6	69.53	-36.0%
3	61.8	62.31	0.8%
4	88.08	62.89	-28.6%
5	66.85	62.21	-6.9%
Average	81.66	68.06	-16.6%

This table shows that our distributed approach to the minimum energy broadcast tree solution performed well in comparison to the centralized approach, and actually outperformed BIP on average. This seems suprising, but the reason for this is due to the order of node addition to the tree structure. In BIP, this order is purely based on incremental cost. In our implementation, we emulate this process by delaying the action of nodes based upon the link cost. However, since the cost at the lowest transmit power is 20 mA, we set the delay to be proportional to the difference from 20 mA in the implementation. An unintended but beneficial consequence of this is that nodes which are two-hops downstream select their parent earlier in the process, allowing nodes which are one-hop downstream to use this information in their parent selection process.

To further validate the results of our test, we computed the total energy cost of each file distribution to compare to the efficiency we observed in our simulations. We logged every transmission, along with the size in bytes of the packet and the transmission power. We used this to compute the total energy cost of the distribution, including the tree construction phase and any recovery messages, in milliampere seconds. We also computed the total energy cost that would have resulted from a blind flood, so that we could compute the EER (Expended Energy Ratio) as we did for the simulations. The average degree of connectivity of the tests was around 9. Based on the low density graph in Figure 4, the EER for a broadcast session of 27 packets was about 10.03. Our observed results from the implementation showed an average EER of 9.27, which may be expected from a slightly denser network.

4 Conclusions

Through simulation and experimentation 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. Previous approaches 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. Liang, W.: Constructing minimum-energy broadcast trees in wireless ad hoc networks. In: MobiHoc, ACM (2002) 112–122

2. Zagalj, M., Hubaux, J.P., Enz, C.C.: Minimum-energy broadcast in all-wireless networks: : Np-completeness and distribution issues. In: MobiCom, ACM (2002)
3. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: INFOCOM. (2000)
4. Cheng, M.X., Sun, J., Min, M., Du, D.Z.: Energy-efficient broadcast and multicast routing in ad hoc wireless networks. In: IPCCC. (April 2003) 87–94
5. Wieselthier, J., Nguyen, G., Ephremides, A.: Distributed algorithms for energy-efficient broadcasting in ad hoc networks. In: MILCOM. Volume 2. (Oct. 2002)
6. Ingelrest, F., Simplot-Ryl, D.: Localized broadcast incremental power protocol for wireless ad hoc networks. In: ISCC, IEEE Computer Society (2005) 28–33
7. Souryal, M., Klein-Berndt, L., Miller, L., Moayeri, N.: Link assessment in an indoor 802.11 network. In: WCNC. Volume 3. (0-0 2006) 1402–1407
8. Erdogan, S.Z., Hussain, S.: Using received signal strength variation for energy efficient data dissemination in wireless sensor networks. In: DEXA. (2007)
9. Cartigny, J., Simplot, D., Stojmenovic, I.: Localized minimum-energy broadcasting in ad-hoc networks. In: INFOCOM, IEEE Communications Society (2003)
10. Cartigny, J., Ingelrest, F., Simplot-Ryl, D., Stojmenovic, I.: Localized lmst and rng based minimum-energy broadcast protocols in ad hoc networks. In Ad Hoc Networks **3**(1) (2005) 1–16
11. Toussaint, G.T.: The relative neighbourhood graph of a finite planar set. In Pattern Recognition **12**(4) (1980) 261–268
12. Li, N., Hou, J., Sha, L.: Design and analysis of an mst-based topology control algorithm. In IEEE Transactions on Wireless Communications **4**(3) (May 2005) 1195–1206
13. Ingelrest, F., Simplot-Ryl, D., Stojmenovic, I.: Target transmission radius over lmst for energy-efficient broadcast protocol in ad hoc networks. In: ICC. Volume 7. (June 2004) 4044–4049
14. Cheng, M.X., Sun, J., Min, M., Li, Y., Wu, W.: Energy-efficient broadcast and multicast routing in multihop ad hoc wireless networks. In Wireless Communications and Mobile Computing **6** (2006) 213–223
15. Wan, P.J., Calinescu, G., Li, X., Frieder, O.: Minimum-energy broadcast routing in static ad hoc wireless networks. In: INFOCOM. (2001) 1162–1171
16. Wieselthier, J., Nguyen, G., Ephremides, A.: The energy efficiency of distributed algorithms for broadcasting in ad hoc networks. In: WPMC. Volume 2. (Oct. 2002)
17. Srinivasan, K., Levis, P.: Rssi is under appreciated. In: EmNets. (May 2006)
18. Reis, C., Mahajan, R., Rodrig, M., Wetherall, D., Zahorjan, J.: Measurement-based models of delivery and interference in static wireless networks. In Rizzo, L., Anderson, T.E., McKeown, N., eds.: SIGCOMM, ACM (2006) 51–62
19. Barr, R., Haas, Z.: Jist/swans website (April 2004)
20. Barr, R., Haas, Z.J., van Renesse, R.: Jist: an efficient approach to simulation using virtual machines. In Software - Practice and Experience **35**(6) (2005) 539–576
21. : Crossbow technology: Low-power wireless mote solutions overview. `http://www.xbow.com/Products/wproductsoverview.aspx` (2009)