5-31-2005

Energy Balanced Broadcasting through Delayed Intelligence

Michael R. Gosnell
Missouri University of Science and Technology, mrghx4@mst.edu

Ryan Arbarelli

Maggie Xiaoyan Cheng
Missouri University of Science and Technology, chengm@mst.edu

Bruce M. McMillin
Missouri University of Science and Technology, ff@mst.edu

Recommended Citation

M. R. Gosnell et al., "Energy Balanced Broadcasting through Delayed Intelligence," *International Conference on Information Technology: Coding and Computing*, vol. II, Institute of Electrical and Electronics Engineers (IEEE), May 2005.
The definitive version is available at https://doi.org/10.1109/ITCC.2005.144

Energy Balanced Broadcasting Through Delayed Intelligence

Michael R. Gosnell   Ryan Albarelli   Maggie X. Cheng   Bruce McMillin
{mrghx4, rma, chengm, ff}@umr.edu
UMR Computer Science, 1870 Miner Circle, Rolla, MO 65409–0350
Phone: (573) 341–4491   Fax: (573) 341–4501

Abstract— **Ad hoc wireless networks are growing in popularity and usefulness, however they rely on broadcasting as a fundamental process for routing. Improvements to broadcasting have made ad hoc networks more feasible, but sometimes benefit only specific situations. Delayed Intelligence (DI) is proposed as a new load balancing approach where small delays are introduced to allow distributed responsibility delegation. Preliminary results show delayed intelligence, when applied in existing broadcasting methods such as passive clustering, can be used to improve the energy disparity and therefore extend ad hoc network lifetime.**

Keywords: Wireless Ad Hoc Network, Broadcasting, On-Demand Routing, Load Balancing, Energy Conservation.

I. INTRODUCTION

Computational devices have grown smaller, more powerful, and more functional over recent years. The introduction of communication capabilities in such devices only drives them to become more commonplace. With the disconnection of devices from wired networks, old methodology and practices are often not adequate to handle wireless networking needs. Traditional wired routing protocols can frequently transmit routes among routing devices with small overhead. Wired routers draw power from a virtually unlimited source, and have an abundance of available bandwidth. Wireless networks, on the other hand, are often constrained by substantially lower bandwidths and limited power. These differences have driven research in wireless ad hoc networks consisting of no infrastructure.

In ad hoc networks, network-wide broadcasting (simply referred to as broadcasting) is one of many functions that requires special attention when transitioning from wired mediums. The lack of any set topology or continuous connections between hosts yields the *broadcast storm problem* [1]. Specifically, concerns regarding redundant broadcasts, contention, and collision are problematic, and referred to as the *broadcast storm problem*. As broadcasts are generally responsible for route finding and potentially for maintenance in unicast and multicast routing protocols [2], a more efficient broadcasting algorithm can significantly aid an ad hoc network.

Section II briefly describes some of the most widely recognized broadcasting algorithms and how they relate to the broadcast storm problem. Passive clustering is described in Section III as one potential algorithm for improving broadcasting within wireless ad hoc networks. Delayed intelligence is introduced and described in Section IV as a way to enhance broadcasting and obtain better utilization of available resources. Analysis of passive clustering with delayed intelligence and initial simulation results are presented in Sections V and VI. Conclusions and future research are addressed in Section VII.

II. BROADCASTING ALGORITHMS

Global dissemination, or broadcasting, of information in a wireless ad hoc network presents a host of problems. The disconnected nature of an ad hoc network means each node has a fixed power consumption limit. As most power consumption goes to transmitting and the ideal capacities of transmission channels are significantly lower than the raw channel bandwidth [3], eliminating excess communication is essential.

Excess communication can stem from redundant broadcasting as well as collisions, both of which are components of the *broadcast storm problem* [1]. Redundant broadcasts are most prevalent in basic (or blind) flooding (Section II-A). Contention is experienced when nodes in close proximity transmit at the same time, contending for the shared transmission medium. Reducing redundant broadcasts helps reduce contention, but redundancy is not the only cause of contentions. Collisions occur when multiple message transmissions overlap a reception area as in the *hidden terminal* [4] or *hidden node* [2] problem. By reducing excess broadcasts, collisions and contentions are decreased while energy is conserved.

Efficient flooding requires removal of duplicate, unnecessary broadcasts. Two helpful methods often implemented in layers underneath networking include *jitter* and *Random Assessment Delay* (RAD) [2]. Jitter is a very small random delay introduced as part of some broadcasting algorithms retransmission process. All nodes receive an initial transmission at nearly the same point in time. Jitter delay attempts to account for the fact that multiple nodes could attempt a retransmission of that data at the same point in time. By varying the initial sending time slightly, contention is reduced and some collisions can be avoided.

Random assessment delay is a random-length delay that is generally longer than jitter. The RAD is introduced in some broadcasting algorithms' nodes where there is a local decision as to whether to retransmit received data. RAD is the amount of time a node waits while listening for duplicate transmissions. RAD can be used in the algorithms from Sections II-B and II-C.

The delaying processes are somewhat comparable to the exponential backoff delaying used when collisions occur under Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [5]. The concept of slight delays for increased efficiency is of key importance and is the foundation of our delayed intelligence enhancements presented in Section IV.

Before introductions of broadcasting algorithms some terminology will be defined. n will denote any arbitrary node in an ad hoc network. $neighbor(n)$ denotes some neighbor of n. $\{neighbor(n)\}$ is the set of all neighbors of n.

A. Flooding

The simplest way to propagate a message through a wireless ad hoc network is to have each node transmit the message once. This is known as *blind flooding* and generally has the highest coverage percentages. This algorithm is one of a few known ways to obtain effective coverage in highly mobile or low density networks [1], [2]. However, with high message traffic or areas of high density, performance suffers (mainly due to channel contention) [1], [2].

Jitter, or some other delaying mechanism, is essential for cutting down contention when blind flooding is used. Even with jitter, blind flooding still has the most overhead and waste of resources with respect to the broadcast storm problem.

B. Probabilistic Algorithms

On average, not all nodes in an ad hoc network will need to retransmit a broadcast for every node to receive the message. If a node n transmits a message out to radius r, every node with distance from n less than or equal to r will also receive the message. This concept is termed the *wireless multicast advantage*, and refers to the byproduct of each $neighbor(n)$ within r also receiving n's transmission [6]. Probabilistic broadcast algorithms use this property to limit nodes from retransmitting received broadcast messages. Using some defined probability p in a probabilistic-based broadcasting algorithm, a node will retransmit each unique received broadcast with probability p.

Using $p = 100\%$ will equate to the blind flooding algorithm discussed previously. Conversely, setting p to a value too small can yield situations where the broadcast does not get transmitted throughout the entire network. Probabilistic broadcasting algorithms might be better suited than blind flooding, however, much of the performance of probabilistic broadcasting algorithms relies on p and the relationship of p to the network density and mobility. Without *a priori* network topology, probabilistic algorithms risk excessive message sending or poor performance in low density areas.

C. Counter Algorithms

Similar to a probabilistic approach is a counter-based broadcasting algorithm. In counter algorithms, nodes are set with some count C which acts as a threshold for determining retransmissions of received broadcasts. When n's broadcast message is initially received at some $neighbor(n)$, it will calculate a RAD and wait for duplicate transmissions. If duplicate broadcasts are detected, a new RAD is calculated and the internal message count is increased. If the internal message count reaches C before the RAD is reached, the message is not transmitted from $neighbor(n)$. However, if the number of duplicate messages is below the threshold count C by the time the delay has expired, $neighbor(n)$ will transmit the message.

This scheme will essentially mimic the probability of sending with a probability dynamically based on a node's local density. If there is a dense area, fewer nodes will need to transmit because some of them will reach C. In sparse areas, a higher relative percentage of the nodes will participate and help insure better coverage. Results from [1] suggest a $C \geq 3$ effectively covers the general ad hoc cases and $C = 6$ roughly equals the performance of blind flooding.

D. Distance and Location Based Algorithms

Distance and location-based broadcasting algorithms reduce the number of broadcasts by estimating or computing the additional coverage provided by each transmission. In the distance-based broadcasting algorithms, starting from a broadcast at some node n, the neighboring nodes $\{neighbor(n)\}$ estimate their distance from n using received signal strength and initial transmission power. During each $neighbor(n)$'s RAD, duplicate broadcasts are detected and the minimum distance is retained from all equivalent messages. At the end of the RAD, if the stored minimum distance is less than some predefined distance threshold D, the transmission at $neighbor(n)$ is not performed. Thus, if $neighbor(n)$ senses that some n close by has already transmitted the message, it can assume the additional coverage will be insignificant and will not transmit.

Location-based broadcasting algorithms are inherently more beneficial than distance-based because they utilize specific location knowledge to more accurately determine additional coverage. With GPS (or other position-based) knowledge of n and $neighbor(n)$, additional coverage can be calculated in a manner similar to the distance-based algorithms. With position knowledge, each $neighbor(n)$ has the added benefit of being able to calculate if they fall completely within previous coverage areas and what additional area would be reached with a retransmission.

Given that a broadcast's retransmission can achieve a maximum additional coverage area of 61%[1] [1], distance and location-based algorithms stress the importance of maximizing the effectiveness of subsequent transmissions. Localized location-based algorithms could include location information during normal transmissions at some small additional cost per message. Distance methods could utilize received signal strength to estimate distance between nodes where no additional message traffic is needed. Both methods allow for good all-around broadcasting algorithms with the knowledge that any subsequent unicast or multicast routes might be error-prone in mobile environments.

[1] Assuming omnidirectional antennas and ideal unobstructed transmission radii

E. Virtual Infrastructure Algorithms

Another method for modeling transmissions and determining which nodes retransmit a broadcast relies on using geometric properties. One geometric representation uses two-dimensional disks with radii corresponding to the broadcasting power of the nodes. If all nodes transmit at the same power, then it is a unit disk graph. Based on these geometric properties, virtual infrastructures such as clusters ([7]–[9]), connected dominating sets ([2], [10], [11]), or trees ([6], [12]–[14]) can be constructed, in which only a specific subset of nodes (gateways and clusterheads, dominating nodes, or non-leaf nodes) need to retransmit.

In clustering, each cluster of nodes is ruled by a single node, usually deemed a *clusterhead*. The clusterhead is responsible for broadcasting traffic to all nodes within its cluster. Clusterhead nodes are never 1-hop away from one another, but can be varying distances apart depending on which construction algorithm is used. Generally, nodes used to connect clusterheads are *border* or *gateway* nodes. The final state typically used in clustering algorithms consists of some *ordinary* node which does not participate in retransmissions of broadcasts.

Localized clustering algorithms are generally built on 1-hop and 2-hop neighbor knowledge. Performance is again constrained by persistent message overhead, allowing neighbor knowledge to remain current. Collision and contention are observed most drastically under high traffic and in highly mobile networks where the neighbor knowledge can not easily be kept current. Passive clustering, discussed in more detail throughout Section III, maintains clusters without requiring complete neighbor knowledge.

III. PASSIVE CLUSTERING

Passive Clustering (PC) is an algorithm designed to supplement ad hoc routing algorithms that construct clusters in which only a subset of nodes will be responsible for broadcasting. As described in [7]–[9], passive clustering is a completely passive protocol which constructs soft-clusters on the fly by attaching node state information onto existing message traffic. No maintenance messages are required to determine, advertise, or update cluster information as is required by many of the clustering, location based, and connected dominating set algorithms.

Passive clustering creates soft-clusters by determining *clusterhead* nodes without complete neighbor knowledge. Recall from Section II-E that a clusterhead is a node responsible for forwarding messages to all of its neighbors. All 1-hop neighbors of a clusterhead can not be clusterheads themselves. *Gateway* nodes link multiple clusterheads together. Passive clustering allows for multiple gateway nodes and is described in more detail later. If a node is neither a gateway nor clusterhead, it does not need to retransmit broadcasts and is classified as an *ordinary* node. A final node state in passive clustering is termed *initial* and is the starting state as well as the state nodes revert to if no traffic has been seen for some defined timeout period. More complete rules for the alteration of node states is given in [8] [2], however the basic definition of passive clustering should be enough to understand the effects of such a scheme.

With passive clustering, clusterheads are selected using the *first declaration wins* principle. The first declaration wins principle dictates that the first node to broadcast itself as clusterhead automatically becomes the clusterhead. All other nodes within the broadcasting radius of the first clusterhead-declaration broadcast must eventually declare themselves as gateway or ordinary nodes, depending on subsequent node declarations.

For the *gateway selection heuristic*, a node monitors the number of neighboring clusterheads (NC) and neighboring gateways (NG). With a predefined gateway redundancy coefficient $\alpha \mid \alpha \geq 0$ and a gateway redundancy factor $\beta \mid \beta \geq 0$, a node can declare itself a gateway when [8]

$$\alpha \times NC + \beta > NG \tag{1}$$

It is pointed out that α and β can be local parameters, unique to each node, that can be adjusted based on density, channel usage, etc.

The gateway selection heuristic is designed to limit the number of gateway nodes that link clusterheads together. Limiting the number of gateways can save additional message overhead but still allow for redundancy. Redundancy is especially helpful in the cases of higher node mobility and unreliability.

IV. DELAYED INTELLIGENCE

With high transmission costs of ad hoc networks, low-overhead communication is ideal. Assuming all nodes that receive a wireless transmission do so at about the same time, delaying a retransmission for a length of time (much greater than the network propagation delay) could relay information without adding additional messages. For an example, assume a basic flooding algorithm was implemented where each node would delay retransmissions one second for each percentage of local energy depleted. Following an initial transmission from some n, each $neighbor(n)$'s transmission would indirectly inform n of the remaining energy of $neigbor(n)$. Clearly, an implementation with such large delays is not desirable, however the concept of delaying communications for a net gain is worthwhile. The proposed Delayed Intelligence (DI) is such a concept, in which a node will delay its response for retransmission according to the received signal power and its local remaining energy.

To maximize efficiency of transmissions using DI, nodes will be penalized proportionally to the power of received broadcasts. In dense networks, the physically close neighbors would receive and record high transmission power. As in distance-based broadcasting algorithms, these nodes would not cover as many new nodes as those nodes toward the edge

---

[2] For example, if a node marked as clusterhead receives a message directly from a neighboring clusterhead (a situation that may occur with node movement), it will transition to an ordinary node.

of the broadcasting radius of the initiating node, and will be penalized most heavily before responding.

With wireless nodes cooperating in an ad hoc environment, each should equally share the responsibilities of broadcasting while attempting to keep all nodes powered. To maximize longevity, each node will also be penalized with a delay inversely proportional to its local remaining energy. Thus, the higher the local energy in a node, the higher the probability that node will respond first and undertake a broadcasting role.

V. APPLYING DELAYED INTELLIGENCE IN PASSIVE CLUSTERING

Passive clustering was chosen as a typical example to show the benefit of delayed intelligence in reducing excessive transmissions in broadcasting. With the *first declaration wins* principle in passive clustering, the nodes allow for indirect weighing in selection of clusterhead and gateway nodes. By not immediately responding in a scenario where a node could declare itself as a clusterhead or gateway, the probability that neighboring nodes become dominant within the cluster increases. By utilizing delaying properties with respect to received signal power and local remaining energy, nodes using Passive Clustering with Delayed Intelligence (PCDI) can indirectly communicate information about their appropriateness to be dominant within clusters.

It is important to note that the delaying is only introduced when a node is preparing to send a message which would change its state to clusterhead or gateway. If the node is ordinary or already in one of the aforementioned states, no delays are introduced. The metric proposed for each node's wait time W is calculated as

$$W = \delta_1 \times \frac{receivedPower}{localEnergy} \tag{2}$$

or alternatively using distance

$$W = \delta_2 \times \frac{1}{receivedDistance \times localEnergy} \tag{3}$$

where δ_1, δ_2 are constants scaling to other parameters used in PCDI.

For PCDI to be beneficial at balancing energy consumption among nodes, the nodes must revert to their *initial* state from time to time. Thus, either the timeout value must be set sufficiently low, the network broadcasting must be sufficiently infrequent, or the nodes must have a mechanism to reset to *initial*. Relying on a low timeout value or consistently low network traffic is not reasonable. Therefore, PCDI is configured to periodically reset nodes to their *initial* state.

Upon declaration of *clusterhead* or *gateway* status, a node will revert to *initial* when it has consumed some predefined percentage of its remaining energy. For example, if a node with 100% energy declares itself a clusterhead with a energy depletion threshold of 75%, it will revert to *initial* at 75% energy. Whereas a node declaring itself a clusterhead with 50% remaining energy will revert to *initial* at 37.5% energy (75% of its initial energy level of 50%). This method gives more frequent opportunities to hand off broadcasting responsibility as a node's energy becomes depleted. Of course, if the normal timeout is reached before the energy depletion threshold is reached, the status is set to *initial* and the energy threshold is discarded.

VI. SIMULATION

As a proof of concept of delayed intelligence, a simple time-stepped simulation environment was constructed where Passive Clustering with Delayed Intelligence (PCDI) could easily be compared against passive clustering[3] and blind flooding. The simulation environment represented a user-defined arbitrarily-sized, two-dimensional space containing any number of specified nodes uniformly distributed at random coordinates. Without loss of generality, we will call distances $Marks(M)$. The area of the simulations was set to 25, 50, or 100 square $Marks$ and the number of nodes varied between 25, 50, and 100. Although the conditions may not adequately simulate what one would consider realistic values, distances could be scaled up proportionally to simulate more realistic conditions as addressed in Section VII.

Each node was initialized with an equal amount of energy and equal transmission power in all directions. The sending capabilities of all nodes was 20 $Marks$. The starting energy for each node was set to 100% under all tests. Without loss of generality, the initial energy reserve of each node can be called 100 $Watts$. Energy parameters were evaluated from [3], [7], [15]–[17], and 0.04 $Watts$ and 0.01 $Watts$ were chosen as the respective power requirements for sends and receives. During each timestep, a node would randomly send a broadcast message with a 4% probability. The timeout period used in PC and PCDI varied over 10, 20, and 30 timesteps (ts). The δ used in calculating delaying wait times scaled W from 0 to 10 timesteps. The goal of these simulations was to determine if energy consumption could be balanced using delayed intelligence.

A. Density Variations

For each parameter set, 100 simulations were run. Fig. 1 shows the average energy remaining in simulated environments at the time the first node depleted its energy. The results are compared over the network density D defined as

$$D = \frac{TotalNodes}{EnvironmentArea} \tag{4}$$

As one might expect, PCDI obtains lower remaining average energy in dense networks, where more nodes can be selected to share the burden of communication.

B. Timeout Values

The relationship between PC and PCDI timeout values and remaining energy is shown in Fig. 2. Generally, lower timeout values produced more even energy consumption in both passive clustering and passive clustering with delayed intelligence. This indicates that more frequent opportunities

---

[3] Passive Clustering was implemented with the gateway selection heuristic with $\alpha = 1$, $\beta = 1$.

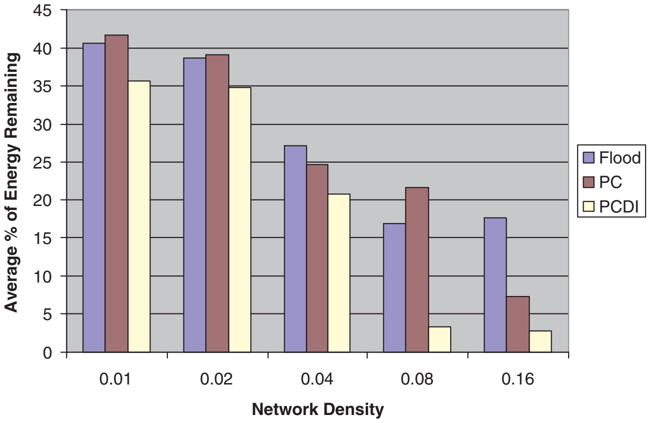

Fig. 1. Comparison of remaining energy at the time the first node in the system depleted its energy. Node density was adjusted by varying the testing dimensions between squares of 25, 50, and 100 $Marks$ with 25, 50, or 100 nodes.

for broadcasting responsibility transfer can even out energy consumption through delayed intelligence.

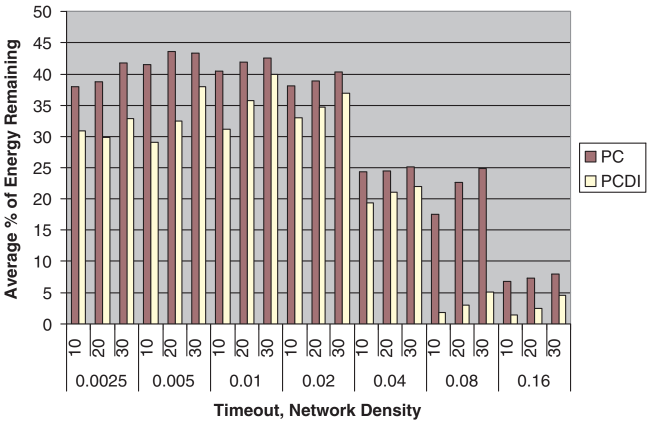

Fig. 2. Comparison of remaining energy at the time the first node in the system depleted its energy. Trials were varied over timeout values of 10, 20, and 30 timesteps (ts) for PC and PCDI. Results were compiled over areas of $(25M)^2$, $(50M)^2$, $(100M)^2$, using 25, 50, and 100 node runs for each timeout value.

C. PCDI Energy Threshold

Recall from Section IV that PCDI incorporates a *energy depletion threshold*, or simply a energy threshold. When a node changes its state to gateway or clusterhead, the local energy level is recorded and a threshold value is set at a predefined percentage lower than the initial value. If the energy threshold is reached before the node relinquishes its gateway or clusterhead state, the node is forced to transition to its initial state. The energy threshold attempts to account for cases where heavy traffic maintains a cluster so long that delayed intelligence is not able to effectively distribute the broadcasting load among nodes. As shown in Fig. 3, energy threshold choice produces no significant variations to average remaining energy when the message traffic is held at a 4% message initiation rate.

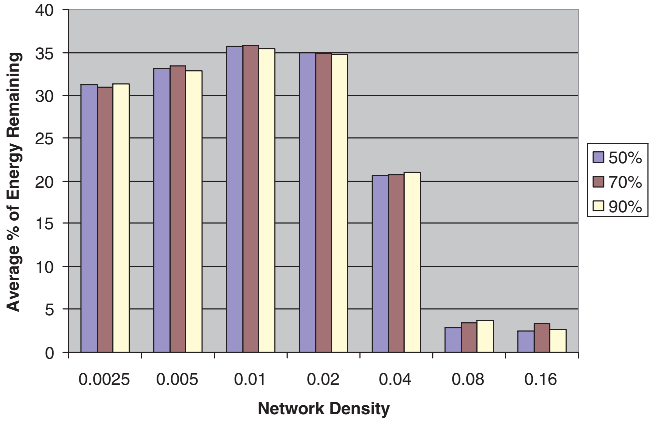

Fig. 3. Comparison of PCDI over different energy threshold values. The energy threshold was the percentage of initial energy at which point a clusterhead would revert to *initial* status if it had not previously done so.

Further analysis of the energy threshold over varying network traffic conditions is presented in Table I. Although a energy threshold value of 50% is better than 90% and 70% in low traffic situations, it appears to suffer under high traffic. This may be due to less frequent opportunities to hand off broadcasting duties toward the end of a nodes life. Holding the timeout value at 10, shown in Table II, verifies that more frequent opportunity for responsibility sharing can yield more balanced energy usage at a lower energy threshold. A more thorough analysis of energy thresholds including the possibility of local, dynamically calculated thresholds may be reasonable, but is left for future work.

TABLE I

COMPARISON OF ENERGY THRESHOLD'S EFFECT ON AVERAGE REMAINING ENERGY IN PCDI OVER VARYING TRAFFIC CONDITIONS. TESTS WERE RUN IN A $(25M)^2$ GRID WITH 25 NODES ($0.04\ \frac{Nodes}{M^2}$).

Probability of sending	Threshold 50%	70%	90%
20%	26.7%	19.8%	18.2%
10%	9.2%	9.8%	11.1%
5%	3.7%	4.5%	4.3%
4%	3.2%	3.2%	3.1%

VII. CONCLUSIONS AND FUTURE WORK

With the introduction of slight delays and an energy threshold timeout, delayed intelligence is effective in balancing energy consumption over various simulated network conditions. Delayed intelligence is the process by which a node will delay its response for retransmission according to the received signal power and its local remaining energy. Delayed intelligence is most noticeable under high densities, where more nodes

TABLE II

COMPARISON OF ENERGY THRESHOLD'S EFFECT ON AVERAGE REMAINING ENERGY IN PCDI OVER VARYING TRAFFIC CONDITIONS WHERE THE TIMEOUT VALUE WAS HELD AT 10. TESTS WERE RUN IN A $(25M)^2$ GRID WITH 25 NODES ($0.04\ \frac{Nodes}{M^2}$).

Probability of sending	Threshold		
	50%	70%	90%
20%	9.1%	9.2%	10.8%
10%	4.0%	4.5%	4.0%
5%	2.4%	2.5%	2.3%
4%	1.9%	1.9%	2.0%

are available to share broadcasting responsibilities. PCDI was compared to passive clustering and blind flooding and had lower average energy remaining per node under all tested densities. This indicates energy disparity is less severe in a PCDI enhanced broadcasting system.

An additional note to consider with location-based measurements concerns mobility. Recall from Section IV, one of the goals of delayed intelligence is to extend the distance between clusterhead and gateway nodes. If the majority of nodes used for broadcasting are far apart and the respective path is used as a unicast route, disconnected routes could become frequent problems.

Future goals include testing using NS2 [18] with a full protocol implementation of PCDI over on-demand routing protocols. In NS2, mobility and power usage can be simulated under more realistic network conditions. Additionally, concerns with disconnected routes can be monitored and our protocol could be expanded accordingly.

Since passive clustering is designed to extend an existing protocol in the formation of clusters, improvements to an on-demand protocol, such as AODV [4], would also benefit with PCDI. Future work includes analysis of AODV enhancements and possible combination of an enhanced AODV protocol with PCDI.

REFERENCES

[1] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, 2002.

[2] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM Press, 2002, pp. 194–205.

[3] Y. Lu, W. Wang, Y. Zhong, and B. Bhargava, "Study of distance vector routing protocols for mobile ad hoc networks," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*. IEEE Computer Society, 2003, p. 187.

[4] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," February 1999, pp. 90–100. [Online]. Available: http://www.cs.ucsb.edu/∼ebelding/txt/aodv.ps

[5] "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications," in *IEEE Std 802.3*. IEEE Standard for Information technology, March 2002. [Online]. Available: http://standards.ieee.org/getieee802/802.3.html

[6] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, march 2000, pp. 585–594.

[7] T. J. Kwon and M. Gerla, "Efficient flooding with passive clustering (PC) in ad hoc networks," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 1, pp. 44–56, 2002.

[8] M. Gerla, T. Kwon, and G. Pei, "On demand routing in large ad hoc wireless networks with passive clustering," in *Proceedings of IEEE WCNC*, September 2000. [Online]. Available: http://www.cs.ucla.edu/NRL/wireless/

[9] Y. Yi and M. Gerla, "Scalable AODV with efficient flooding based on on-demand clustering," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 98–99, 2002.

[10] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.

[11] X. Li and I. Stojmenovic, "Broadcasting and topology control in wireless ad hoc networks," in *Handbook of Algorithms for Mobile and Wireless Networking and Computing*. CRC Press, July 2004. [Online]. Available: http://www.site.uottawa.ca/%7Eivan/broadcasting.html

[12] F. Ingelrest and D. Simplot-Ryl, "Localized broadcast incremental power protocol for wireless ad hoc networks," INRIA, France, Tech. Rep. RT-290, January 2004. [Online]. Available: http://www.lifl.fr/RD2P/Papers/rd2p.bib/ingelrest-tr-04

[13] M. Cagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proceedings of the 8th annual international conference on Mobile computing and networking*. ACM Press, 2002, pp. 172–182.

[14] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM Press, 2002, pp. 112–122.

[15] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM Press, 2001, pp. 70–84.

[16] A. Safwat, H. Hassanein, and H. Mouftah, "Energy-aware routing in MANETs: analysis and enhancements," in *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*. ACM Press, 2002, pp. 46–53.

[17] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[18] *The ns Manual*, UC Berkeley, LBL, USC/ISI, and Xerox PARC, December 2003. [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html