A Fuzzy-Based Hierarchical Energy Efficient Routing Protocol for Large Scale Mobile Ad Hoc Networks (FEER)

Wasim El-Hajj, Dionysios Kountanis, Ala Al-Fuqaha, Mohsen Guizani
Department of Computer Science, Western Michigan University, MI
4601 Campus Drive, Kalamazoo, MI 49008
E-mail: welhajj@cs.wmich.edu

***Abstract*— A mobile Ad-Hoc network (MANET) is a collection of autonomous arbitrarily located wireless mobile hosts, in which an infrastructure is absent. In this paper we propose a fuzzy-based hierarchical energy efficient routing scheme (FEER) for large scale mobile ad-hoc networks that aims to maximize the network's lifetime. Each node in the network is characterized by its residual energy, traffic, and mobility. We develop a fuzzy logic controller that combines these parameters, keeping in mind the synergy between them. The value obtained, indicates the importance of a node and it is used in network formation and maintenance. We compare our approach to another energy efficient hierarchical protocol based on the dominating set (DS) idea. Our simulation shows that our design out performs the DS approach in prolonging the network lifetime.**

***Index Terms*—mobile wireless ad-hoc network, hierarchical design, energy efficiency, and fuzzy logic.**

I. INTRODUCTION

An Ad-Hoc network is a collection of autonomous arbitrarily located wireless hosts (also called nodes), in which an infrastructure is absent. Two nodes can communicate directly with each other if they are within each others' range; otherwise, intermediate nodes have to relay messages for them. Therefore, each node in such a network must provide services such as routing, address assignment, DNS-like name translation, and more.

As the number of nodes in an Ad-Hoc network becomes large, the overhead in computing, storing, and communicating routing information becomes prohibitive. It has been proven [1, 2] that a flat network has poor scalability. In [1], theoretical analysis show that the node throughput declines rapidly to zero as the number of nodes in the network increases. It was also shown, that even when the nodes are optimally placed, the network of size N can not provide a per-node throughput of more than $c/\sqrt{N}$ bits/sec, where c is a constant. Gupta et al. [2] reports the experimental results of the scaling law described in [1] employing IEEE 802.11 technologies. The results show that the decline in throughput is like $c/N^{1.68}$ bits/sec, which is considerably worse than the theoretical results.

In addition to the scalability problem of the flat network, each node has a big energy limitation. Usually, Ad-Hoc nodes die fast because they have limited energy batteries. In this paper, we propose a fuzzy-based energy efficient hierarchical network design approach to overcome the scalability and the limited energy problem that AD Hoc wireless nodes have in a flat network.

The remainder of this paper is organized as follows. Section II includes some related work done in energy efficient routing in Ad-Hoc networks. Section III describes the system model we use. Section IV discusses the fuzzy logic controller. Section V includes a detailed explanation of our approach. Section VI presents simulation results of our approach and its effectiveness compared to the DS approach. Section VII concludes the paper and discusses future work.

II. RELATED WORK

Researchers have proposed many hierarchical clustering algorithms that differ in their objective. One of these objectives is energy efficient clustering [3-13]. Since nodes in an Ad-Hoc network have limited residual energy, an energy efficient hierarchical protocol that aims to maximize the network lifetime becomes crucial. In this section, several hierarchical energy efficient protocols are described.

MINPOW [17] minimizes the total power consumption (for communication) on a route. It is essentially the distributed Bellman-Ford algorithm with sequence numbers where the cost is the total power consumption instead of the hop count. MINPOW has a big disadvantage in the sense that it only relies on the link cost to determine the route. A better approach is to take into account the link cost and the node cost.

In [10], the authors proposed an energy efficient clustering based on the dominating set (DS) marking algorithm [13]. The network is represented as a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. The DS is a set D of vertices of G such that every vertex of G is either in D or adjacent to a vertex in D. Nodes in the DS are considered gateways (cluster heads) and other nodes in the network join the gateways creating the clusters. The DS nodes should be connected in order to enable the routing of messages between clusters. Since finding the DS is NP-complete, Wu et al. [13] proposed a simple distributed algorithm that marks a node as a gateway if two of its neighbors are not directly connected. To route traffic from a source to a destination, the source sends the traffic to its gateway, the gateway routes the traffic to the destination gateway, and then from the destination gateway to the destination node.

The major problem related to this approach is the way the original DS is calculated. The DS is generated based on the node degree and its connectivity with its neighbors. Residual energy is not considered as one of the deciding factors when finding the DS nodes. So, the resultant DS (cluster heads) might be composed of nodes that have low residual energy while other nodes in the network have high residual energy. The main objective of the DS approach is to minimize the DS updates rather than to balance the energy consumption among all mobile nodes.

Our approach differs from other approaches in: (1) Using fuzzy logic to aggregate the residual energy, traffic, and mobility parameters. (2) Designing a fault tolerant backbone. (3) Designing energy efficient routing that maximizes network lifetime.

III. SYSTEM MODEL

In our work, we assume a model where the radio dissipates more energy while transmitting than receiving. Each node has a battery with limited residual energy (RE). Each node, equipped with antennas, can control its transmission power (power level). The higher the power level, the more distance a node covers and the more energy it consumes. The lifetime of a node depends on: (1) the traffic load the node is routing, (2) the energy consumed while transmitting or receiving the traffic load, and (3) the residual energy on the node. These parameters should satisfy the following inequality:

$$Load\,(b/s) \times Communication(J/b) \times t(s) : \; RE(J) \qquad (1)$$

Where, $Load$ is the amount of traffic passing through a node in bits per second (b/s), $Communication$ is the amount of energy dissipated by the node when transmitting, receiving, or both in joules per bit (J/b), and RE is its residual energy in joules (J). Let t be the lifetime of the node in seconds (s), then the above inequality can be rewritten as:

$$t(s) \le RE(J) / Load(b/s) \times Communication(J/b) \qquad (2)$$

i.e., the node dies when the energy consumed by communication exceeds its own residual energy. The equation used to calculate the energy consumed when a node communicates is given by:

$$E_{consumed} = (packetLength / bitRate) \times P_c \qquad (3)$$

Where, P_c is the power consumed if the node is transmitting a packet, receiving it, or both.

IV. FUZZY LOGIC CONTROLLER

One of the major steps in our hierarchical network design is to select nodes that act as cluster heads (CH). Many factors affect the choice of a CH. The CH should be able to handle the traffic generated to/from its cluster nodes. Therefore, it should have high residual energy. Also, the CH should not be too mobile because this leads to high packet loss rate. There is a correlation between the values of these parameters (residual energy, traffic, mobility). Because these parameters have different units and their values can be defined in ranges, fuzzy logic is used to express the effect of their interaction. Two major steps are needed to develop the fuzzy logic controller: (1) define member functions for each input/output parameter and (2) design the fuzzy rules. The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs, define functional overlap between inputs, and determines an output response. The rules use the input membership values as weighting factors to determine their influence on the output sets. The membership functions, discussed below, were designed to satisfy the following two conditions: (1) Each membership function overlaps only with the closest neighboring membership functions; (2) for any possible input data, its membership values in all the relevant fuzzy sets should sum to 1 (or nearly so). The membership functions are as follows:

- Residual energy is represented by 3 triangular membership functions as shown in Figure 1. The triangular membership function is specified by three parameters $\{a, b, c\}$ as follows:

$$triangular(x : a, b, c) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \le x \le b \\ (c-x)/(c-b) & b \le x \le c \\ 0 & x > c \end{cases} \qquad (3)$$

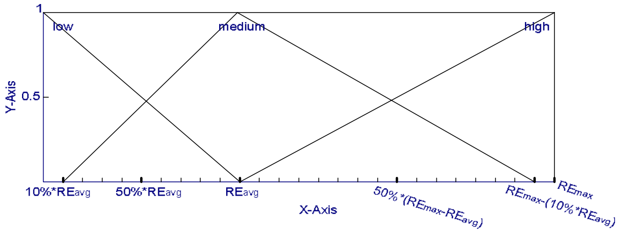

Figure 1: Three RE membership functions representing low, medium, and high RE.

The x-axis represents the value of the residual energy in Joules. The three triangular membership functions representing the RE are marked by low, medium, and high. The average RE (RE_{avg}) of all network nodes is calculated and is considered to be the center of the medium range. Nodes with REs less than RE_{avg} are classified as medium and/or low, while those larger than RE_{avg} are classified as medium and/or high.

- Mobility is represented by 2 trapezoidal membership functions as shown in Figure 2. A Trapezoidal membership function is specified by four parameters $\{a,b,c,d\}$ as follows:

$$trapezoid(x:a,b,c,d)=\begin{cases}0 & x<a\\ (x-a)/(b-a) & a\le x<b\\ 1 & b\le x<c\\ (d-x)/(d-c) & c\le x<d\\ 0 & x\ge d\end{cases} \quad (4)$$

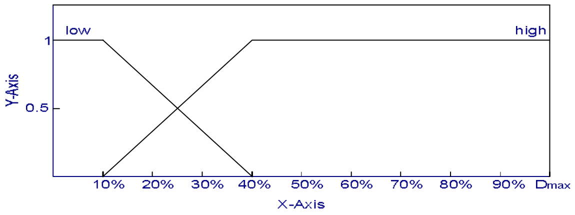

Figure 2: Two mobility membership functions representing low and high mobility.

The mobility is measured by the change in the average received signal strength (RSS) between the node and its neighbors as it moves from one location to another. Each node translates received signal strengths (RSS) to distances. Then, the average distance is calculated. A new average is taken after the node moves to a new location. The difference between both averages is fed to the fuzzy logic controller. Notice that the difference computed can not be more than the distance a node can cover when it uses its highest power level. Therefore, the maximum distance is represented by D_{max} which is the distance covered when a node uses its maximum power level (P_{max}). The mobility model is designed to assign low mobility to nodes that have less than 10% difference and high mobility to nodes that have greater than 40% difference. A transition between low and high mobility occurs at 25% difference.

- Traffic is represented by 3 triangular membership functions as shown in Figure 3.

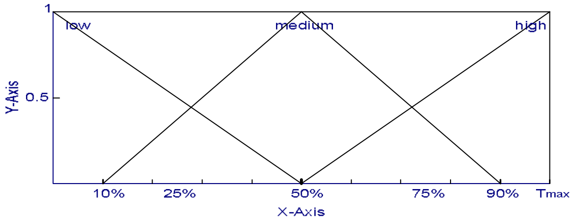

Figure 3: Three traffic membership functions representing low, medium, and high traffic

The x-axis indicates the normalized value of the input traffic load. The max bit rate is considered to be 11Mbps (when using 802.11g). The three triangular functions determine a smooth transition between low, medium, and high traffic.

- The output is represented by three trapezoidal member functions as shown in Figure 4. They indicate whether the output is good, acceptable, or bad. The value of the output is a number between 0 and O_{max}. The smooth transition from bad and acceptable occurs between 5% and 45%, while the smooth transition from acceptable to good occurs between 55% and 95%.

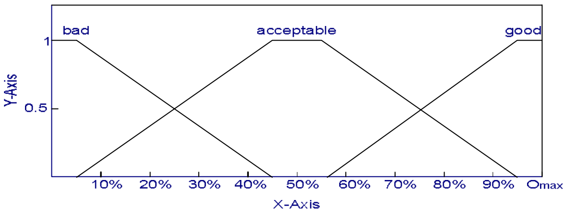

Figure 4: Three output membership functions indicating whether the aggregated weight is bad, acceptable, or good.

Six fuzzy rules are used: (1) if RE is high, Output is good. (2) If mobility is high OR traffic is high, Output is bad. (3) If RE is medium AND mobility is low, Output is acceptable. (4) If RE is low, Output is bad. (5) If RE is medium AND traffic is medium, Output is acceptable. (6) If RE is medium AND traffic is low, Output is acceptable. The above rules summarize the CH properties, i.e. the CH is preferred to have high RE, low mobility, and low traffic.

V. PROPOSED APPROACH

Each node in the network passes its parameters (RE, traffic, and mobility) to the fuzzy logic controller. The controller returns a weight representing these parameters. A centralized node is used to make the initial configuration of the network. Then, the network configuration is broadcasted to all the nodes in the network. The central processing node (CPN) is considered to be the node with the lowest id (in future work, a simple protocol will be used to elect the node with the highest RE as the CPN).

The fuzzy-based hierarchical energy efficient clustering protocol (FEER) can be divided into four parts. (1) Elect nodes to act as CHs. (2) Associate each node in the network with a CH. (3) Introduce a network recovery approach to ensure a fault tolerant backbone. (4) Design an energy efficient routing between nodes.

1. CH choice: Given a random configuration of an Ad-Hoc network, the first task is to adjust the power level of each node in order to get a connected network. The algorithm starts with all the nodes using their minimum power level. If node *A* is reachable by node *B*, a communication link is created between *A* and *B*. The topology created is tested for connectivity using the "disjoint set union" algorithm which is almost linear. The algorithm runs in $O(n\alpha)$, where *n* is the number of nodes in the network and α is a small number ($\alpha \leq 5$ for all $n \leq 2^{2^{16}}$).If the network is not connected, the power level at each node is

incremented and the process is repeated until a connected network is established (if there is one). Note that the power level at each node will be lowered as much as possible after constructing the hierarchical structure.

Having established a connected network, we propose a simple algorithm to pick the cluster heads. Since the CHs form the backbone network, they must be connected. The node with the highest aggregated weight (W: the output of the fuzzy logic controller) is chosen to be the first cluster head. The neighbors of the first CH are stored in vector *Ne*. From the nodes in *Ne*, the node with the max weight is chosen to be the next CH (the chosen CH is supposed to cover new nodes). New neighbors are added to vector *Ne*. The above step is repeated until all the network nodes are covered. After each iteration, the elected CHs are removed from *Ne*. In the worst case, the algorithm needs to visit all n nodes in the network. Therefore, its complexity is $O(n)$. The advantage of this approach is that there is no need to know the number of CHs a priori. The CHs are chosen automatically by the algorithm.

2. Creating the clusters: After the selection of CHs, each node associates itself with a CH to form a cluster. Nodes in a cluster are of two types: one-hop nodes and two-hop nodes. One-hop nodes are the direct neighbors of the CH. Two-hop nodes, also called guests, are nodes that can reach the cluster head through a one-hop neighbor.

The idea of a guest node is introduced in [15]. One reason for using guest nodes is that 1-hop clustering schemes form a highly overlapping cluster structure with a large number of small clusters. Such a structure may cause difficulties in the channel spatial reuse and thus leading to low network capacity. So, using cluster guests reduces the number of clusters. Another reason for using cluster guests is to avoid the ripple effect (re-clustering the whole network from scratch). A mobile node that moves out of the CH range can join a close cluster as a guest rather than re-clustering.

Nodes connect to the CH with the maximum lifetime (Equation 2). A node might have the ability to reach more than one CH using 1-hop or 2-hops. The lifetime of each CH is calculated and the one with the maximum lifetime is chosen. The algorithm connects nodes having higher traffic first.

If the chosen CH was 1-hop away from the node, the node connects directly to that CH and becomes part of the cluster. If the chosen CH was 2-hops away from the node, another node (connector) is needed to connect the node to the CH. Thus, the node and the CH should share at least one neighbor. The neighbor with maximum lifetime is chosen to act as a connector. Note that the lifetime of the nodes is calculated considering whether the node is: transmitting, receiving, or both transmitting and receiving. A node not assigned to a cluster is connected to a neighboring CH with maximum lifetime.

At this point each node is associated with a cluster. Nodes can not communicate directly unless they go through the CHs they are connected to. The above algorithm achieves a hierarchical structure that maximizes the network lifetime, because nodes with minimum lifetime are avoided.

The power of this algorithm lies in the fact that the traffic demand and mobility are integrated within the algorithm and the clusters are chosen to best handle the traffic demand. The algorithm also determines the traffic load that each CH needs to handle. These loads are routed through other CHs in order to reach the destination. But, what if the channel capacity between CHs can not handle those loads? Or, what if one of the links was down because of the channel impairment? These issues can be solved by providing the recovery algorithm (discussed next).

3. Network Recovery: Having designed an energy efficient hierarchical structure, it is important to make the network resilient to link failure and that the CHs can handle the traffic flow. To check for resiliency and traffic demand, the min cut and the max flow algorithms are used (defined later). The min cut in the network should be greater than or equal to the network fault tolerance requirement (M) and the maximum flow should be satisfied. If one or both conditions fail, appropriate recovery techniques are deployed. The network created by CHs is represented by a weighted graph $G = (V, E)$, where V is the set of vertices and E is the set of edges and the weight on each edge represents its capacity.

Min Cut recovery: The min cut of a graph G is the minimum number of edges needed to disconnect the graph. A simple randomized min-cut algorithm is used to find the min cut. The probability of finding the min cut in the first round is $2/n(n-1)$. The algorithm can be repeated many times (n^2 is an appropriate number) in order to increase the probability of getting the correct min cut. The complexity of the min cut algorithm is $O(n)$. The total complexity depends on how many times the algorithm is repeated.

If the min cut was greater than or equal to 2 (M=2), then the network is considered to be reliable and the max flow is checked. If the min cut was equal to one, one link failure can bring the network down. Two recovery approaches can be used to increase the min cut and thus increase the reliability of the network. The first approach is an iterative approach that increases the number of cluster heads *without* increasing the power level. Suppose that cluster heads A and B are connected by a single link, the approach finds all the neighbors of A and B and chooses the neighbor with the maximum lifetime as a new CH. This algorithm is repeated for any two CHs having a single link between them. Thus, more paths are generated between CHs and fault tolerance is increased. If the first approach was not able to increase the min cut, the power level of each node is increased and the whole process is repeated.

Max flow recovery: If the min cut condition was satisfied (min cut > 1), a max flow algorithm is applied between every *(s, d)* pairs to check if the traffic flow can be handled by the links. The algorithm used is the Edmonds-Karp algorithm and it runs in $O(ve^2)$. If the max flow is satisfied, the network hierarchical structure becomes complete; else the second recovery method is used.

4. Routing: In the formation of the network topology, every node was assigned the same power level. When the design is completed, every node (which is not a CH) that belongs to a cluster lowers its power level as much as possible, provided that it can still reach its CH and its connector (if it exists). The routing is composed of three steps: (1) if the source is not a CH, it sends its message to its representative CH, else it does nothing. (2) Now the CH acts as the new source. The *k*-shortest path algorithm is used to get different paths from the source CH to the destination CH. Each path is assigned a cost which is equivalent to the minimum lifetime of a node across the path. The path with the max cost (max lifetime) is used to route the traffic to the destination. Along the path, each node updates its lifetime. (3) If the destination CH is not the final destination, it relays the message to the final destination node in its cluster.

VI. SIMULATION RESULTS

In this section, we evaluate the performance of FEER via simulation. We compare our approach to that of the dominating set [10]. Node mobility is simulated using Gauss-Markov Mobility Model [16] and traffic generation is simulated using the Poisson distribution (exponential inter-arrival time and exponential holding time). Each node in the network can transmit data using 6 different power levels. The higher the power level the more energy the antenna needs to transmit the signal and the more distance the signal can reach (Table 1). The power consumed by the battery when transmitting data is 1749 mW and when receiving is 930 mW. The battery of each node has energy up to 10000 Joules.

Power Level	Transmit Power (dBm)	Receive Power (dBm)	Distance (m)
6	20	-70	302
5	18	-70	240
4	17	-70	170
3	15	-70	135
2	13	-70	107
1	10	-70	76

Table 1: Nodes' power level properties

Figure 5 shows a network of 40 nodes designed using FEER. The black nodes are CHs. Each cluster is composed of the CH and the nodes associated with it. These nodes can be connected to a CH using 1-hop or 2- hop nodes. 9 nodes are elected to act as CHs.

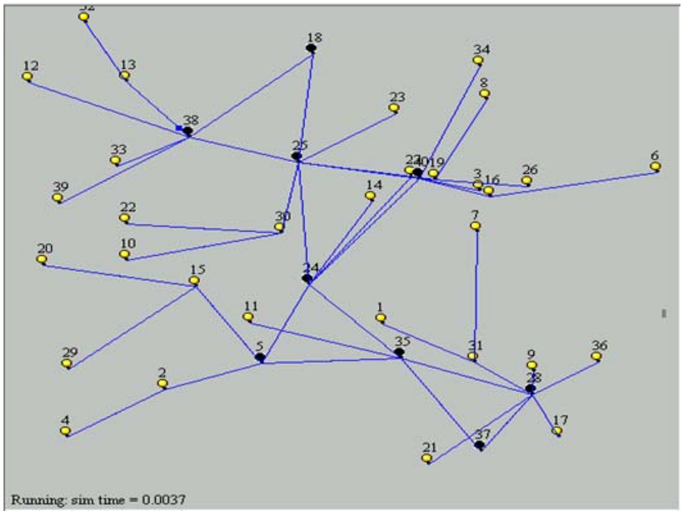

Figure 5: Network topology using FEER

Figure 6 shows the design of the same network using the DS approach. The DS approach elects 25 nodes to act as CHs which is almost 3 times more that FEER.

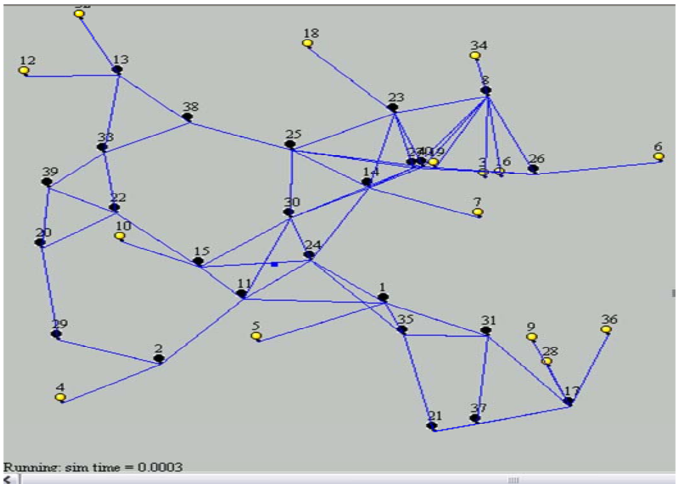

Figure 6: Network topology using DS approach.

The first experiment we conducted varies the network size and measures the network lifetime. Each node sends packets at a rate up to 500kbps. Each packet ranges in length from 50 bytes to 2400 byte. The energy consumed at each node is calculated using Equation 3 (discussed in section III). Figure 7 shows that as the network size increases, FEER produces longer lifetime than the DS approach. Note that networks with large sizes have low lifetimes. This is true because in large networks, the cluster heads will be representing a larger number of nodes and thus their energy is quickly depleted. Using FEER, the lifetime of the network is prolonged by almost 20%. Note that in order to speed up the process of collecting results, we lowered the RE of each node.

Figure 8 shows the number of elected cluster heads using FEER and the DS approach. FEER chooses far fewer cluster heads than the other approach and still improves the network lifetime. For instance, when the network size is 100, FEER elects 24 cluster heads while the DS approach elects 74.

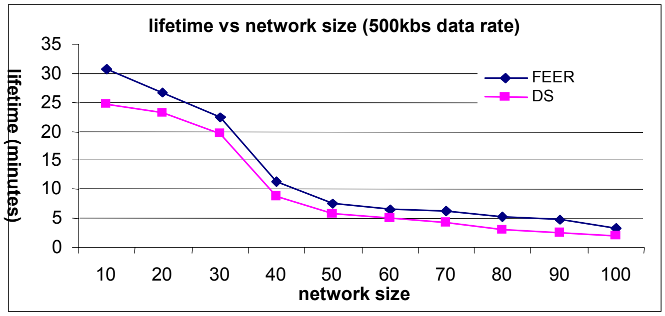

Figure 7: Network lifetime vs. network size. A maximum bit rate of 500kbps is used at each node.

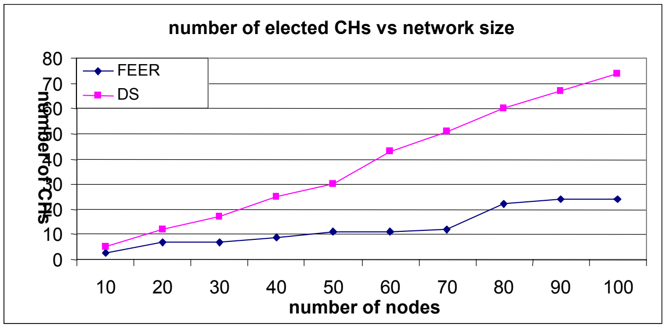

Figure 8: Number of CHs vs. network size

Another experiment was conducted to check the lifetime of the network under heavy traffic. A network of 100 nodes is tested under different data rates that range from 50kbps to 500kbps. Figure 9 shows that as the data rate increases, FEER still achieves longer lifetime (20% better) than the DS approach. Both approaches though feature a decreasing lifetime as the data rate increases.

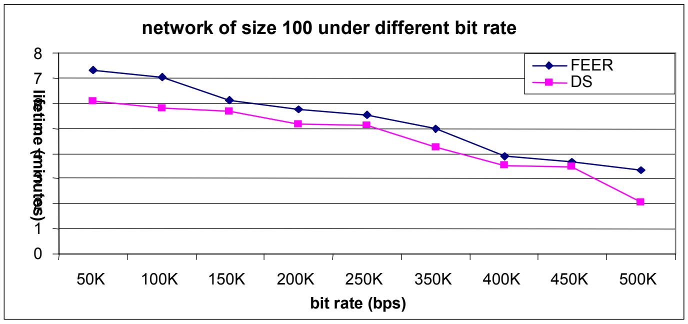

Figure 9: Network lifetime vs. bit rate. The experiment is made on a network of 100 nodes.

VII. CONCLUSION

In this paper, we proposed a fuzzy-based energy efficient hierarchical network design approach that prolongs the lifetime of a mobile Ad Hoc network. A fuzzy logic controller was developed to aggregate the important parameters that characterize a wireless node. These parameters include residual energy, traffic, and mobility. The aggregated weight is used to elect cluster heads. The other nodes of the network were assigned to cluster heads such that the network lifetime is prolonged. Simulation results show that FEER designs networks that have 20% more lifetime than the DS energy efficient protocol. Future work includes designing a simple protocol to elect the central processing node (CPN) and designing an efficient cluster maintenance algorithm.

REFERENCES

[1] P. Gupta, and P. R. Kumar, "The Capacity of Wireless Networks", IEEE Transactions on Information Theory, vol. IT-46, no. 2, pp. 388-404, March 2000.

[2] P. Gupta, R. Gray, and P. R. Kumar, “An Experimental Scaling Law for Ad Hoc Networks,” May 16, 2001. http://black1.csl.uiuc.edu/~prkumar/

[3] CR Lin, M. Gerla, “Adaptive Clustering for Mobile Wireless Networks”, IEEE JSAC, vol. 15, pp. 1265-1275, Sept. 1997.

[4] D. Amis and R. Prakash, “Load-Balancing Clusters in Wireless Ad Hoc Networks,” *Proc. 3rd IEEE ASSET’00*, Mar. 2000, pp. 25–32.

[5] J.-H. Ryu, S. Song, and D.-H. Cho, “New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks,” *Proc. IEEE ICC’01*, vo1. 3, June 2001, pp. 862–66.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," *Proceeding of the Hawaii International Conference System Sciences,* Hawaii, January 2000.

[7] M. Younis, M. Youssef and K. Arisha, “Energy-Aware Routing in Cluster-Based Sensor Networks”, *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002),* Fort Worth, TX, October 2002.

[8] A. Manjeshwar and D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks,” *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing,* San Francisco, CA, April 2001.

[9] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems," *Proceedings of the IEEE Aerospace Conference,* Big Sky, Montana, March 2002.

[10] J. Wu *et al.*, “On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks,” *J. Commun. and Networks*, vol. 4, no. 1, Mar. 2002, pp. 59–70.

[11] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems," *Proceedings of the IEEE Aerospace Conference,* Big Sky, Montana, March 2002.

[12] S. Lindsey, C. S. Raghavendra and K. Sivalingam, "Data Gathering in Sensor Networks using the Energy*Delay Metric", *Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing,* San Francisco, CA, April 2001.

[13] J. Wu and H. L. Li, “On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks,” *Proc. 3rd Int’l. Wksp. Discrete Algorithms and Methods for Mobile Comp. and Commun.*, 1999, pp. 7–14.

[14] M. Veyseh, B. Wei and N. F. Mir, "An Information Management Protocol to Control Routing and Clustering in Sensor Networks", Journal of Computing and Information Technology- vol. 13, no. 1, Mar. 2005

[15] J. Y. Yu and P. H. J. Chong, “3hBAC (3-hop between Adjacent Clusterheads): a Novel Non-overlapping Clustering Algorithm for Mobile Ad Hoc Networks,” *Proc. IEEE Pacrim’03*, vol. 1, Aug. 2003, pp. 318–21.

[16] V. Tolety. Load reduction in ad hoc networks using mobile servers. Master’s thesis, Colorado School of Mines, 1999.

[17] V. Kawadia and P. R. Kumar, “Power control and clustering in ad hoc networks”. *Infocom*, 2003.