Flow Driven Routing in Ad Hoc Sensor Networks

Hamid Mousavi
CE Department,
Sharif University of Tech.
Tehran, Iran.
h_mousavi@ce.sharif.edu

Hesam ChiniForooshan
CE Department,
Sharif University of Tech.
Tehran, Iran.
chineeforoushan@ce.sharif.edu

Mohammad E. Rafiei
CE Department,
Sharif University of Tech.
Tehran, Iran.
rafiei@ce.sharif.edu

Mohammad Ghodsi
CE Department,
Sharif University of Tech.
Tehran, Iran.
ghodsi@sharif.ir

ABSTRACT
Ad hoc sensor networks are usually composed of a large number of tiny sensors that have limited capabilities. One of the important problems in these networks is how the packets are routed to their destinations. The answer to this question should be provided locally in each node for the sake of energy conservation and scalability. However, we can not always make the best decision without any global information of the network. Considering these, we propose a Flow Driven Routing method which inspiring by the maximum flow concept, estimates the global status of the network. The global information leads to a better decision making, and the experimental results show that the total number of packets that can be routed is increased by 8.1% with respect to one of the best existing methods. This value for the networks with big holes and lower density can reach up to 17%.

Keywords
Ad hoc sensor networks, routing, maximum flow.

1. Introduction
In recent years, apparent improvements in micro-electro mechanical technology have led to creation of tiny sensors, which has caused an impressive enhancement in the field of sensor networks. Typically, ad hoc sensor networks consist of a large number of tiny wireless sensors with no predefined infrastructure. Each of these sensors usually has limited energy supply, restricted communication and sensing range, and low processing capability. Sensors are usually considered to be static or with little mobility. Having only one destination is also of usual assumptions.

Ah hoc sensor networks are being used in many applications. Environmental monitoring or controlling, security, military, and context-aware computing are a few examples of these applications.

One of the most important issues in ad hoc sensor networks, like other types of networks, is the routing algorithms [4]. In such networks, an effective routing algorithm is the one which makes the lifetimes of the networks as long as possible. The lifetime of a network can be defined to be the first time that the energy supply of a sensor depletes [18]. There can also be other definitions for the networks' lifetimes specially based on the time that the network is partitioned. To increase the networks' lifetimes and also to make the algorithm scalable, we have to make the routing decisions locally. On the other hand, with global information of the networks, we can obviously make a better decision for routing. To resolve this contradiction, local operations should be used to estimate the global information, so decisions can be made based on these global information locally.

Another important characteristic of a good routing algorithm is the effective use of the local information and usual assumptions in ad hoc sensor networks. Two of the most important local parameters of a node are its location and energy and those of its neighbors. Considering the static nature of the sensors, these parameters, specially positions of the neighbors, can be obtained more easily using some approaches such as those in [14,15,16].

Several methods have been proposed for routing algorithm in ad hoc networks, some of which have been designed for mobile ad hoc networks (MANET) [11,12,13]. Most of these algorithms can easily be used in ad hoc sensor networks too. However, the more the method is specialized, the better the performance will be. There are also methods that designed specifically for ad hoc sensor networks [1,2,3,6,10,17]. Unfortunately, most of these approaches either have used weak estimation of global information or have not used any estimation at all.

In this paper we propose a geographical and energy-aware routing based on the idea of maximum flows in networks. The maximum flow problem is concerned with finding the maximum amount of data (flow) that can be sent from a source node toward a destination node (sink) in a directed weighted graph. Weight of each edge in this graph is known as the capacity of that edge which restricts the amount of data which can be forwarded through that edge at a time. Obviously, computing the exact value of maximum flow is not feasible in such networks. Hence, considering the existence of only one destination called Base Station (BS), we estimate this value for each node. The main results show 8.1% improvement in total number of packets that can be sent to BS using this type of information with respect to one of the best existing methods. Our approach also decreases the number of dead nodes in about 11.1%. This performance is much better in networks with big holes.

The rest of this paper is organized as follows: related works are summarized in section 2. Section 3 contains some preliminary definitions. In section 4, the proposed algorithm is illustrated precisely, and section 5 shows the experimental results. Some discussions are included in section 6. Finally, conclusion and future work are presented in section 7.

2. RELATED WORK

Karp et al. proposed a new geographical ad-hoc routing called GPSR in [1] in which only local information are used to make forwarding decisions. To forward a packet, if there exists any closer neighbor to the packet's destination than the node itself, GPSR forwards the packet to the closest neighbor to the destination. Otherwise, if the packet is reached to a region in which there is no closer neighbor to the destination, GPSR routes the packet around the perimeter of the region. GPSR can be used in networks which nodes are mobile, and it guarantees no loop occurrence in routing packets. However, GPSR has some shortcomings too. First of all, it dose not use the amount of node's energy in its forwarding decision makings. In addition, nodes are assumed to operate in promiscuous listening mode and consequently consume energy.

In [2], Yu et al. proposed geographical and energy aware routing called GEAR. The main goal in GEAR is to disseminate a packet in a specific region. So, the algorithm is twofold: forwarding the packet toward the region and disseminating it in the region recursively. In GEAR, each node has learned cost for each target region. Every time a node has to forward a packet to a neighbor that is farther from destination, it will update its learned cost for that region to which the packet is supposed to be sent. This way, next time the packet will not be sent to this node more probably because of node's higher learned cost. After routing the packet toward the target region, GEAR recursively disseminates the packet inside the region.

There are also works which are focused on specific types of networks. For example in [6], an energy-aware multi-path routing in wireless sensor networks is proposed, but the approach works only for some specific network topologies such as grid and torus. Some theoretical points on capacity limits of sensor networks with grid or torus topologies are proposed in [7]. Servetto et al. in [5] use random graph concept to construct a multi-path routing algorithm in large-scale wireless sensor networks. Their work should also be considered specified for structured networks since some information of networks are used which can not be easily computed in networks with no infrastructure.

Brown et al. theoretically explore a new routing objective for energy aware ad-hoc networks [8]. Using maximum flow concept and linear programming, and considering that a good estimation of total data flows is known, they tried to prolong the connectivity of networks. However, their model seems not to be practicable enough. Considering that all sources and destinations and also the information that the sources will generate, are known, an approach in [9] is introduced based on maximum flows and augmenting path techniques. However, most of these researches are too theoretical.

3. PRELIMINARY DEFINITIONS

As mentioned, we consider that there is only one destination called *base station* (BS). This is not a constraining assumption since it is common to have only one destination in ad hoc sensor networks. We also consider all sensors to be immobile and to have equal sensing areas.

This section is devoted to clarify some notions and definitions which are frequently used throughout the paper. In the paper, we may use node to mean a sensor. Each sensor (node) in the network may have some neighbors. Actually, two nodes are neighbors if and only if each of them is located in the sensing area of the other one. A neighbor of a node that is closer to BS than the node itself is called its *child*. Similarly, neighbors of a node which are farther from BS than the node itself are called its *parents*.

We define a *path* as a sequence of nodes in which consecutive nodes are neighbors and no node can occur more than once in the sequence. The first and last nodes in a path are called *source* and *destination*, respectively. We will call a path destined to BS to be *non-increasing* if the first node in the path is the parent of the second one, the second one is the parent of the third one, and so on. In other words, if a packet follows a non-increasing path, it gets closer to BS in each step and finally reaches to it. Other paths destined to BS are called *increasing*.

After these preliminary definitions, we can define more important concepts. There are three important parameters that are used to make a better decision in forwarding packets; *direct flow* (DF), *indirect flow from children* (IFC), and *indirect flow from parents* (IFP). These parameters are defined as follows:

Definition: *Direct flow* of node *N* toward BS is the number of bytes which can be routed from *N* to BS using only non-increasing paths from *N* to BS.

Definition: *Indirect flow from children* of node *N* toward BS is the total number of bytes which can be routed from *N* toward BS using the paths which are not non-increasing and start with one of the children of *N* (i.e. the first hop in the path should be one of the children of *N*.)

Definition: *Indirect flow from parents* of node *N* toward BS is the total number of bytes which can be routed from *N* toward BS using the paths which are not non-increasing and start with one of the parents of *N* (i.e. the first hop in the path should be one of the parents of *N*.)

4. FLOW DRIVEN ROUTING (FDR)

The whole idea of routing a packet from its source toward BS can be expressed using the notion of LFD, which stands for *Local Forwarding Decision*. LFD means that each node's decision to select a neighbor as the next-hop of the routing path should be made locally. At each node, this decision is made using a combination function of three major parameters of each neighbor, and selecting the most promising one. These parameters are the amount of the flow that a node can send to BS, its distance to BS, and its current energy.

Through the process of decision making in each nodes, the values of its flows and those of its neighbors will be updated, frequently. This way, next time a packet reaches to the node; it can find a better route to BS using these updated flows. In addition to indicating whether a node has any flow to BS or not, these flows contain the global status of all paths starting at them. However, since we could not compute exact value of a node's flow, we had to divide nodes' flows to three parts (DF, IFC, and IFP.) Routing algorithm and updating approach will demonstrate why we used these types of flows.

As already discussed, there is no easy way to compute the real values of the defined flows in a distributed manner. Restricted features of the sensors make the problem more complex.

Nevertheless, we can approximate these values instead of computing their exact values. Before explaining how to estimate these values, it is better to understand how the FDR works. Hence, we first explain the routing algorithm, and then describe the approach that we use to approximate these parameters at the start of the algorithm. Finally, we will show how to update these parameters.

4.1. Routing Algorithm

FDR always tries to route packets trough non-increasing paths. This has some benefits such as energy conservation, faster routing, and preventing loop occurrence in the routing approach. In the case of knowing that there is no direct or indirect flow from a node's children to BS, FDR tries an increasing path using one of the parents of the node as the next-hop.

Consider that node *N* wants to send packet *P* to BS. If *N* has some direct flow to BS, among all children of *N* which has direct flow to BS, the child *C* which maximizes the following function will be selected:

$$\text{DF\_func}(C) = \alpha\times DF_C + \beta\times E_C - (1-\alpha-\beta)\times D_C, \quad (1)$$

where α and β are tunable weights, DF_C is the direct flow of node *C* normalized by the largest direct flow among all the children of *N*, E_C is the current energy of *C* normalized by the largest energy among all children of *N*, and finally D_C is the distance between C and BS normalized by the maximum such distance among all children of *N*. Informally, according to this metric function, FDR tries to select the closest possible node to BS with higher direct flow and energy.

If *N* does not have any direct flow, its children will have obviously no direct flow either. In this case, if *N* has any indirect flow from its children, among all children of *N* which has indirect flow to BS, the child *C* which maximizes the following function will be selected:

$$\text{IF\_func}(C) = \alpha\times IF_C + \beta\times E_C - (1-\alpha-\beta)\times D_C, \quad (2)$$

where IF_C is the total indirect flow (estimated by $IFC_C + IFP_C$) of *C* and the other parameters are the same as what were defined in the previous paragraph. Note that in this formula we could define deferent values for parameters α and β, but the same values work fine.

The other case is when there is no child that we can send the packet P to (i.e. both DF and IFC are zero in the node.) In this case, if there exists a parent with direct flow, the one which maximizes the function DF_func will be selected. Otherwise, if there exists any parent with indirect flow, the one which maximizes the function IF_func will be chosen as the next-hop. At last, the nodes that have no direct and indirect flow will discard all received packets.

4.2. Estimating the Flows

Knowing how the FDR works, it is much easier to describe the approach that we used to estimate the initial flows. It is worth noting that without any initialization method for the flows, the algorithm still works fine because of the existence of an updating approach. However, in this case, the updating algorithm needs more time to estimate the flows, since there are no initial values for them. This usually causes more energy consumption early in the network's lifetime because of imprecise estimation of the flows which results in weaker routing.

According to the routing algorithm, nodes with positive direct flow do not need to know about their indirect flows. Considering this fact, there is no need to estimate indirect flows of nodes that already have direct flow. Similarly, we also do not need to compute IFP if there exists any IFC for a node. We use the following formula to estimate direct flow of node *N* (DF(*N*)):

$$DF(N) = E_N + \sum_{C\in CH} (DF(C)/|CH_C|), \quad (3)$$

where CH is the set of the children of node *N*, $|CH_C|$ is the number of children of *C*, and E_N is the number of bytes that can be transferred using current energy of *N*. The formula simply says that the direct flow of a node is estimated by the sum of direct flows of its children with some refinements. To approximate the two types of indirect flows (IFC(*N*) and IFP(*N*)) of a node, we use the following to formulas:

$$IFP(N) = \sum_{P\in PA} DF(P), \quad (4)$$

$$IFC(N) = \sum_{C\in CH} IFP(C) + \sum_{C\in CH} IFC(C), \quad (5)$$

$$IFP(N) = \sum_{P\in PA} IFP(P) + \sum_{P\in PA} IFC(P), \quad (6)$$

where *PA* is the set of all parents of *N*. Next few paragraphs demonstrate how these formulas can be used to initialize flows.

The distributed algorithm for computing these flows is not too complicated; at the start of the algorithm, each node will send an initial packet, called *initPack*, to all of its parents after receiving *initPack* from all of its children. *InitPack* includes the node's direct flow divided by the number of its children. Thus, the first nodes that send *initPack* are the ones that have no child. The only remaining thing is that when a node which has no child is not the Base Station, it should send zero as its direct flow to its parents.

By completion of the first part, each node that has no direct flow tries one of the following ways in the same order to find an indirect flow:

1. If there is any parent that has a direct flow, the node will set its IFP to the sum of the direct flows of all its parents (Formula 4).
2. If there is any child that has any sort of indirect flow, the sum of the indirect flows of all children of the node will be selected as its IFC (Formula 5).
3. Finally, if one of the parents of the node has any sort of indirect flow, the sum of the indirect flows of all parents of the node will be selected as its IFP (Formula 6).

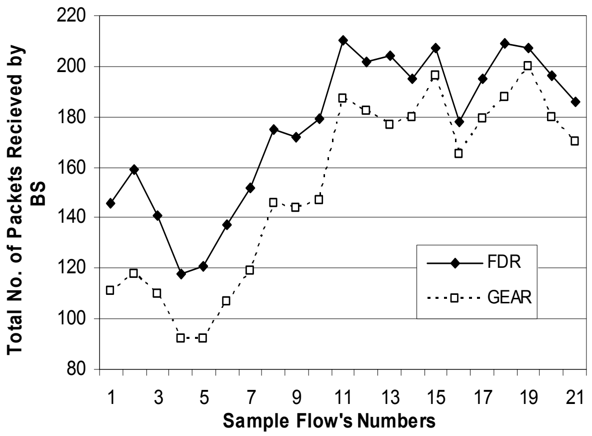
Note that once a node finds any indirect flow, it will stop seeking for more flow. This is necessary to prevent loops.

4.3. Updating the Flows

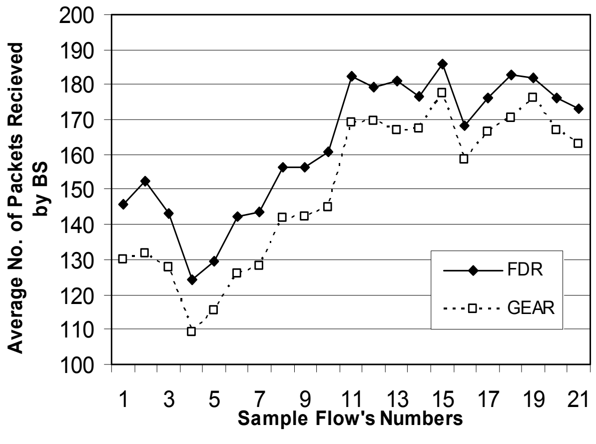
The initially computed flows should be updated when a nodes' energy has been changed because of the routing a packet, sensing, or any other reason. Ideally, when flow of a node varies, all of its parents should be informed. Obviously, this technique is not applicable because of high energy consumption, but the following clues make it applicable:

Table 1: characteristics of the networks constructed for comparison between FDR and GEAR.

Name	Size	Node No.	Disrtibution
WSN0	450m×250m	150	Manual
WSN1	300m×300m	400	Random
WSN2	300m×400m	400	Random
WSN3	400m×400m	400	Random
WSN4	400m×500m	400	Random
WSN5	400m×600m	400	Random



(a) Results for manually constructed network



(b) Total results

Figure 1: The comparison results between FDR and GEAR.

- Each node sends an updating message, called *updatePack*, when it has lost at least T_e units of its direct flow from the last time that it has sent an updating message,
- Each *updatePack* will penetrate at most T levels in the network, and
- Each node adds the changes in its direct flow to the received updating packet before forwarding it to its parent.

We set T_e to be the energy which is need for forwarding ten packets of data (We considered that all data packets has the same size.) We also set T to be 3. More discussions of these parameters are provided in next section. To update indirect flows, we use a simpler approach than what is used for direct flows. Once a node (that has no direct flow) understands that one of its neighbor's flows is changed, it resets the values of IFP and IFC using the same technique which we use to estimate indirect flows in the previous part.

5. EXPERIMENTAL RESULTS

We simulated our proposed routing algorithm using a simple energy model. In this model, all external actions of a sensor including sensing, receiving, and sending a byte needs the same energy independent of distances. We also considered that data packets have the same size and are 20 times larger than other types of packets, i.e. *initPack* and *updatePack*. We also avoid the energy which is needed for computational operations. The sensing area of each sensor is a circle with radius $\sqrt{10}$ meter.

To get a better result, we used different value for α and β in different distances. That is, we used $\alpha = \beta = 0.33$ for distances of farther than 5 hops (i.e. $5 \times \sqrt{10}$ meter) from the Base Station, and $\alpha = \beta = 0.25$ for other distances. That is, the decisions near the Base Station will be made more based on the distance factor than others.

We compared our algorithm with GEAR [2] since it is one of the most well-known geographical and energy-aware routings in sensor networks. As we mentioned before, GEAR has two parts: routing a packet to a node in a region and distributing the packet to all nodes in that region. We only simulated the first part of GEAR with the same energy model since the goal of our routing algorithm is to send a packet to a destination not a region.

To compare our algorithm with GEAR, we construct 6 different networks with different densities. You can find more information on these networks in Table 1. Both FDR and GEAR are tested using 21 random sample flows on these 6 networks. The main results are shown in the Figure 1.

Results indicate the superiority of FDR over GEAR for the networks which are not evenly distributed and have more and bigger holes. Part (a) in Figure 1 shows the comparison between two routing algorithms in a manually constructed network in which 150 nodes are distributed in a 450m×250m area. Low density of this network demonstrates that how big the holes are. Figure 2 shows the distribution of nodes in this network. In such networks, FDR improves the number of routed packets to BS about 17% with respect to GEAR. However, on the average, FDR causes 5% more dead nodes after trying to route sample flows compared to GEAR.

Other networks which we used in order to compare our approach with GEAR are those in which nodes are distributed almost monotonically in the whole area. We construct networks of this type with different densities. Simulation results show that in this type of networks FDR can deliver only 5% more packets compared to GEAR, but FDR conserves much more energy than GEAR. On the average, the number of dead nodes in FDR is 12.5% less than that in GEAR.

Part (b) in Figure 1 shows the total results for both manually and randomly constructed networks. Totally, FDR improves the total number of packets which can be routed to BS up to 8.1% with respect to GEAR. Moreover, on the average, FDR causes 11.1% fewer dead nodes than GEAR after routing the sample flows.

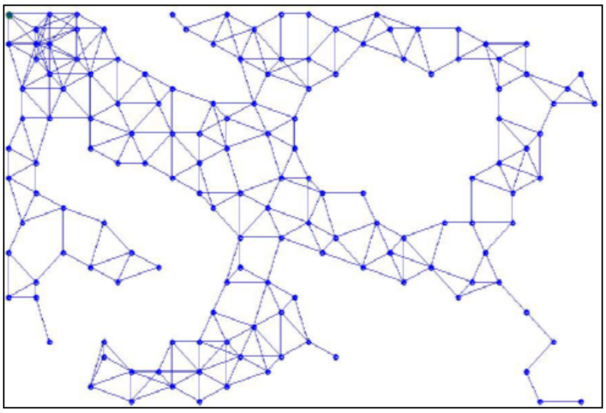


Figure 2: Manually constructed network with big holes and low density. The Base Station is located in the left-most of up-most of the region.

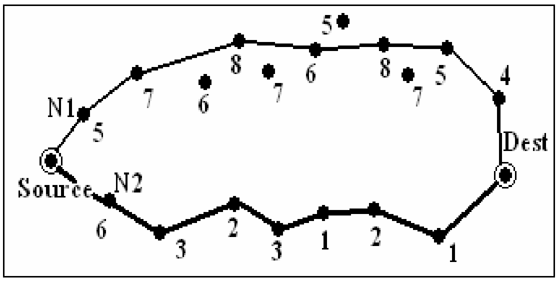


Figure 3: An example of networks with big holes. The number near each node indicates its energy level.

6. DISCUSSION

In FDR, we used some kind of global information of the networks presented as flows. On the other hand, the *learned cost* in GEAR somehow is similar to this global information. However, there are some differences between flows in FDR and learned costs in GEAR, These differences cause FDR makes better forwarding decisions. First of all, flows in FDR will be computed initially, but learned cost in GEAR will be updated only while packets are routed. This can cause more energy consumption for the early packets which should be routed around the holes.

The second difference is the rate of updating. As described, every *updatePack* penetrates T levels (which is greater than one) in the network, That is, we have a faster adaptation. This can lead to better routing around the holes, especially when there are big holes in the network. The third difference is that flows of a node have information about all the paths from the node to the destination while learned cost in GEAR contains only the information of the best path that is computed so far from the node to the destination. So, flows can lead to a better decision making with respect to learned cost.

We should also mention that the bigger the holes in the network are, the better the results of routing will be. One reason is that in such networks, there could be better use of indirect flows since there are more needs to select increasing paths. The other reason is that in these networks deciding that which path around a hole should be selected is a more critical matter.

As an example, Figure 3 shows a simple network with a relatively big hole. Consider the node labeled as *Source* in the figure wants to send a packet to the destination labeled *Dest*. Locally deciding, GEAR chooses neighbor *N2* since it has more energy, and it is also closer to destination than neighbor *N1*. However, looking globally at the network, it can be understood that *N1* is a better choice as the next-hop. This is because of higher energy level of the nodes above the hole with respect to nodes below it. Using flow concept, FDR may more probably choose *N1* as the next-hop. This decision can prolong network lifetime and as a result more packets can be routed to their destinations. Obviously, the same problem can be existed in those networks in which nodes are more evenly distributed. However in such networks for each node, the differences of its neighbors' flows can not be high enough to dictate the correct neighbor to be selected. The reason is that paths destined to BS of these neighbors have too many common nodes, and this makes their flows to be close to each other.

7. CONCLUSION AND FUTURE WORK

Inspired by the network flows concept, we proposed Flow Driven Routing (FDR) which is a geographical and energy-aware routing in ad hoc sensor networks. FDR attempts to make decisions based on global information called flows. These flows are estimated initially, and they are updated through the network's lifetime frequently. The comparison between FDR and GEAR, one of the best-known geographical and energy-aware routing methods, shows the superiority of the FDR over GEAR, especially in those cases that the networks are not evenly distributed and have big holes. The main results are shown in Figure 1. Totally, FDR can send 8.1% more packets than GEAR, and the dead number after routing the sample flows reduces 11.1% in FDR with respect to GEAR.

Because of variety application of ad hoc sensor networks, there are still needs for routing methods with better performance. So, to improve the performance of FDR, a future work can be to investigate approaches which indicate how globally we should decide in forwarding a packet. As an example, a node that is located in highly dense part of a network is better to make its decision locally, that is, only based on the location and energy of its neighbors. Instead, a node that is located beside a hole (like node *Source* in Figure 3) should decide more globally.

8. REFERENCES

[1] B. Karp and H. T. Kung, "Greedy Perimeter stateless routing for wireless network," in *Proc. Sixth annual ACM/IEEE international conference on mobile computing and networking(mobicom)*, Boston, MA, USA, pp.243-254, 2000.

[2] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," *UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023*, May 2001.

[3] B. Krishnamachari, D. Estrin, S. Wicker, "Modeling Data Centric Routing in Wireless Sensor Networks," in the *Proceedings of IEEE INFOCOM,* New York, NY, June 2002.

[4] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Elsevier Ad Hoc Network Journal*, pp. 325-349, 2005.

[5] S. D. Servetto and G. Barrenechea, "Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks," In *Proc 1st ACM Int. Workshop*

on Wireless Sensor Networks and Applications (WSNA), September 2002.

[6] Neha Jain, K. Madathil, P. Agrawal, "Energy Aware Multi-path Routing for Uniform Resource Utilization in Sensor Networks," *IPSN 2003,* pp. 473-487, 2003.

[7] G. Barrenechea, B. Beferull-Lozano, and M. Vetterli, "Lattice Sensor Networks: Capacity Limits, Optimal Routing and Robustness to Failures," In *Proc. of Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, April 2004.

[8] T.X. Brown, H.N. Gabow, and Q. Zhang, "Maximum Flow-Life Curve for a Wireless Ad Hoc Network," In *Proc. of Symposium on Mobile Ad Hoc Networking and Computing,* October, 2001.

[9] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," In *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, pp. 22-31 March 2000.

[10] S. Murthy and J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," In *Mobile Networks and Applications* pp. 183-197, 1996.

[11] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-hocWireless Networks." In *T. Imielinksi and H. Korth, editors, Mobile Computing*, pp. 153-181. Kluwer Academic Publishers, 1996.

[12] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks." In *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, 1998.

[13] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks." In *Proc. of INFOCOM 97*, pp. 1405–1413, April 1997.

[14] S. Capkun, M. Hamdi, and J.P. Hubaux, "GPS-Free Positioning in Mobile ad-hoc Networks," In *Cluster Computing*, 2002.

[15] C. Savarese, J.M. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," In *USENIX Annual Technical Conference*, 2002.

[16] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," In *Proc. IEEE INFOCOM 2004*, March 2004.

[17] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," In *Proc. of the 9th annual international conference on mobile computing and networking*, 2003.

[18] J.H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," In *INFOCOM 2000*, March 2000.