Gossip-based Sleep Protocol (GSP) for Energy Efficient Routing in Wireless Ad Hoc Networks

Xiaobing Hou and David Tipper
Department of Information Science & Telecommunications
University of Pittsburgh
{xiaobing, dtipper}@mail.sis.pitt.edu

***Abstract*— In this paper, we propose a novel energy saving scheme, termed the Gossip-based Sleep Protocol (GSP). With GSP, each node randomly goes to sleep for some time with gossip sleep probability p. When the value of p is small enough, the network stays connected. GSP does not require a wireless node to maintain the states of other nodes. It requires few operations and scales to large networks. We propose two versions of GSP, one for synchronous networks and one for asynchronous networks, and show the advantages of the GSP approach through both simulations and analysis.**

I. INTRODUCTION

In an ad hoc network, mobile nodes must cooperate to dynamically establish routes using multihop wireless links. There is no stationary infrastructure, and each node acts as a router. A packet may have to be forwarded by a sequence of nodes to reach its destination. Many routing protocols have been proposed to solve the dynamic multihop routing problem in ad hoc networks. There are two general classes of routing protocols for ad hoc networks, proactive routing and reactive routing. Proactive routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. Such protocols are termed proactive because they store route information before it is needed. Proactive protocols suffer the disadvantage of additional control traffic that is needed to continually update stale routing entries. Some of the most popular proactive protocols are DSDV [1], WRP [2], OLSR [3] and FSR [4].

Reactive routing creates and maintains routes only when desired by the source node. Therefore, it's also known as on-demand, source-initiated, or demand-driven routing [5]. When a node requires a route to a destination, it initiates a route discovery process within the network, typically, by some form of flooding. This process is completed once a route is found or all possible route permutations have been determined. Once a route has been established, it is maintained by a route maintenance procedure until either the route is no longer desired or the destination becomes inaccessible. Compared to proactive routing, reactive routing consumes far less bandwidth for maintaining the routing tables at each node when only a small subset of all available routes is in use at any time, at the expense of high overhead and delay in setting up the route. Proposed reactive routing protocols include DSR [6], [7], AODV [8], and TORA [9]. A review of ad hoc routing protocols is given in [5].

Since most mobile hosts are not connected to a power supply and battery replacement is difficult, optimizing energy consumption in these networks has high priority. Conventional ad hoc routing protocols, as introduced above, require all nodes keep listening even if there is no traffic or neighbor nodes are totally redundant for each other. Obviously, this wastes energy and significantly reduces the lifetime of the nodes as well as the network's. In this paper, we propose a method improving the energy efficiency of ad hoc network routing by employing a sleep mode. It can also be used in sensor networks, which can be seen as a special case of ad hoc networks with lower mobility and tighter energy budget. Our design has been driven by the following three goals:

- Simplicity: mobile hosts may have limited computing capability and memory resources. Minimized operation and information maintenance are required.
- Scalability: an ad hoc network could be composed of a great number of nodes.
- Connectivity: network connectivity can keep the path setup delay low.

With these goals in mind, we propose the Gossip-based Sleep Protocol (GSP). The core idea is to employ probabilistic based sleep modes - essentially, tossing a coin to decide whether or not a node should sleep for the next period. We show that with certain value of gossip sleep probability p and under certain topology density, the network is still connected, thus works properly. This basic idea is intuitively proposed for a synchronous network without mobility, e.g., a wireless sensor network. Then we show that it also applies to a mobile network. Furthermore, to remove the synchronization overhead, an asynchronous GSP is proposed.

By introducing sleep mode into the network, the total energy consumption of the network can be reduced and the network lifetime can be prolonged. However, the sleep mode may increase the length and the failure rate of a path. Therefore, simulation is conducted to study the effect of GSP based on the network lifetime, throughput and end-to-end delay.

The remainder of this paper is organized as follows. In section II, we present a brief review of current ad hoc network energy efficient routing protocols. Section III presents our Gossip-based Sleep Protocol (GSP). Section IV presents the results of a simulation study. We conclude our work and point out some possible future work in section V.

II. RELATED WORK

Basically, there are two classes of energy efficient ad hoc and sensor network routing protocols employing a sleep mode in the literature, cluster-based and flat. Both of them achieve energy efficiency by employing different topology management techniques. This section presents a brief review of these two classes of routing to provide a better understanding of the current research issues in this area.

In cluster-based routing protocols, all nodes are organized into clusters with one node selected to be cluster-head for each cluster. This cluster-head receives data packets from its members, aggregates them and transmits to a data sink. In some cluster-based routing protocols, the cluster-head assigns TDMA slots to its members to schedule the communication and the sleep mode. Low-Energy Adaptive Clustering Hierarchy (LEACH) [10] is designed for proactive sensor networks, in which the nodes periodically switch on their sensors and transmitters, sense the environment and transmit the data. Nodes communicate with their cluster-heads directly and the randomized rotation of the cluster-heads is used to evenly distribute the energy load among the sensors. Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [11] is designed for reactive networks, where the nodes react immediately to sudden changes in the environment. Nodes sense the environment continuously, but send the data to cluster-heads only when some predefined thresholds are reached. Adaptive Periodic Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) protocol [12] combines the features of the above two protocols by modifying TEEN to make it send periodic data. The cluster-based routing protocols can arrange the sleep mode of each node to conserve energy. However, the high complexity and overhead are incurred.

Flat schemes do not maintain the hierarchical structure. SPAN [13] forms a multi-hop forwarding backbone to preserve the original capacity of the network. Other nodes can go to sleep more often to conserve their energy. Backbone functionality is rotated among the nodes to balance the energy consumption. Geographic Adaptive Fidelity (GAF) [14] conserves energy by identifying nodes equivalent from a routing perspective and turning off unnecessary nodes. The network is divided into grids so that all nodes in the adjacent grids can communicate with each other directly. At each point in time, only one node in each grid is active. GPS or other positioning system is required to get the location information for grid formation. Sparse Topology and Energy Management (STEM) [15] exploits a separate paging channel to wake up nodes to trade off setup latency for energy savings. A low duty cycle radio is used to reduce the energy consumption of the paging channel.

Similar to the above protocols, in this paper, we introduce the sleep mode concept into conventional ad hoc routing protocols to trade off network density for energy efficiency. However, compared with other techniques, the one we used is very simple and scalable without maintaining any information except a timer. Our scheme is totally flat and other flat or cluster-based protocols can be used over our scheme to further reduce energy consumption. For example, the awake nodes in our protocol can be grouped into clusters and thus more efficiently utilize their energy.

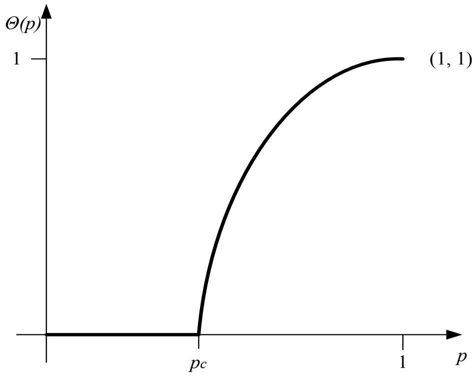


Fig. 1. Percolation probability

III. GOSSIP-BASED SLEEP PROTOCOL

A. Gossip-based ad hoc routing and percolation

In ad hoc networks, gossiping protocols [16] have recently been proposed to reduce the flooding overhead. many ad hoc routing protocols use some kind of flooding scheme to send routing messages. With flooding, every node needs to forward the message once, but this is not necessary since a node with more than one neighbor receives multiple copies of that message. Gossiping reduces this by making some of the nodes discard the message instead of forwarding it. Essentially, a node tosses a coin to decide whether or not to forward the message. The probability p that a node forwards a message is called the gossip probability. Haas, et al., [16] shows that, given a sufficiently large network and a gossip probability p greater than certain threshold, almost all the nodes in the network can receive the message. For example, in a 20×50 grid topology, a value of 0.72 with the first 4 hops from the source node forwarding the message with probability 1 allows almost all the nodes to get the message in almost all the executions of the simulation. This reduces the flooding overhead by about 28%.

The gossiping approach used here implements concepts from percolation theory [17] [18]. In a static infinite network (infinite nodes in an infinite space), if every link or node is open/active with probability p, the network will be grouped into clusters. We are interested in the size and the shape of the clusters as p varies from 0 to 1. The result from percolation theory shows that there exists a critical value $p_c > 0$ such that in the *subcritical phase* (when $p < p_c$), nodes form finite clusters almost surely; in the *supercritical phase* (when $p > p_c$), however, there exist a unique infinite cluster almost surely. The probability that a given node belongs to an infinite cluster $\theta(p)$, termed *percolation probability*, is shown in Fig. 1 [18]. The fraction of nodes belonging to this infinite cluster determines the quality of the connectivity. To date, there is unfortunately no explicit expression of this fraction, nor of p_c. However, we can obtain approximations via simulation, as shown in gossip-based ad hoc routing [16]. Furthermore, we extend this concept to random mobility and an asynchronized scenario.

B. Gossip-based Sleep Protocol (GSP)

As mentioned in section I, the current ad hoc network routing protocols require all the nodes to be awake and keep listening. This wastes a lot of energy, as we will show in section IV. The energy efficient routing protocols and topology management schemes introduced in section II increase the computing complexity and incur extra overhead or equipments. Inspired by gossip-based ad hoc routing, we propose the Gossip-based Sleep Protocol (GSP) to achieve energy efficiency in wireless ad hoc networks without the above drawbacks. Our observation is that if gossiping can make all the nodes receive a message, then the nodes forwarding the message are connected at least by the paths the message passes through. Therefore, in a static network without mobility (e.g., a sensor network), with certain probability p', if gossiping protocols [16] can make almost all nodes in the network receive the message, then if all nodes go to sleep with probability $p = (1 - p')$, almost all the awake nodes stay connected. Thus, we can safely put a percentage (p) of the nodes in sleep mode without losing network connectivity. We term p the *gossip sleep probability*.

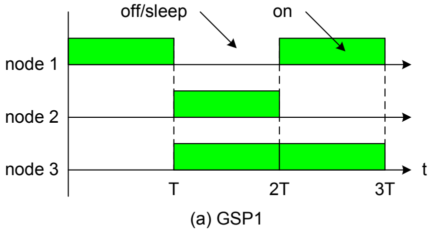
We assume the network is synchronized, i.e., every node decides its own mode for the next period at the same time. The length of the period T is predefined and we term it the *gossip period*. Basically, every node switches on or off based on probability p, which is shown in Fig. 2(a). Although the synchronization is required, the requirement is not strict in case of low mobility (e.g., sensor networks) and it's not necessary to maintain a synchronized clock in every node. The nodes can be synchronized by a control message at the beginning of each period. For example, we can take advantage of the periodic or event-driven ADV(advertisement) message broadcasted from the sink node in Field based Optimal Forwarding [19] without incurring extra overhead. The nodes can also wake up a little bit earlier before the end of each period to wait for the control message and the network performance will not be affected by the extra awake nodes, who are doing nothing but waiting during that short time. The basic version GSP is described as follows and we term it GSP1.

Algorithm 1 (GSP1):

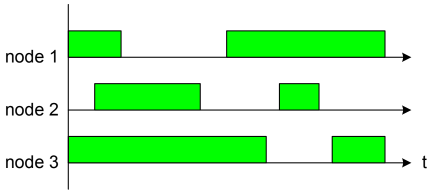
- At the beginning of a gossip period, each node chooses either going to sleep with probability p or staying awake with probability $1 - p$ for this period
- All sleeping nodes wake up at the end of each period
- All nodes repeat the above process for every period

Fairness requires that the length of the period in GSP must be much smaller than the lifetime of the nodes in the network to prevent the condition where a different group of nodes dies in each subsequent period. On the other hand, longer periods avoid frequent link failures.

GSP1 only applies to a network without mobility, such as a sensor network. Two types of network have been studied in [16], regular grid networks and random networks. In this paper, we focus on the random networks since they are more practical. A random network can be constructed by placing nodes randomly in a certain area. In an ad hoc network, random mobility can be viewed as such random placement. At any given instance of time, an ad hoc network with random mobility is a random network or topology, in which GSP1 can be applied. During a period of time, multiple such topologies are generated and make GSP1 work for the whole period. So we can see that although a mobile ad hoc network continuously changes its topology, GSP1 works for the entire lifetime of the network given a random mobility model. Therefore we can extend the GSP1 to the mobile ad hoc networks. Specifically, in an ad hoc network with random mobility, there is a threshold p and if every node goes to sleep for a predefined period with a probability smaller than p, almost all the awake nodes stay connected.



(a) GSP1



(b) GSP2

Fig. 2. Nodes switch on and off in GSP

GSP1 requires all nodes in the network synchronized so that they can toss the coin at the same time. With high mobility, this may either be unachievable or incur overhead and complexity. Since we aim at the applications that require low complexity, a simpler protocol without synchronization is always desirable. To remove synchronization, we assume that every node chooses a uniformly random time interval, termed *gossip interval*, independently and after the time is up, the node will choose another random interval immediately. The nodes in the network do not toss the coin at the same time and none of the nodes has any knowledge of other nodes' coin tossing timing, even its own timing beyond the current interval. Of course, to make it feasible, we assume the possible maximum gossip interval is much smaller than the lifetime of the network. This process is shown in Fig. 2(b). Now, we have two orthogonal dimensions of randomness, space and time. Obviously, at any time instance, the combination of these two kinds of randomness still makes a topology the random topology to which GSP1 can be applied. In other words, in an ad hoc network with random mobility, there is a threshold p and if every node goes to sleep for an independent random interval with a probability smaller than p, almost all the awake nodes stay connected. So we have the asynchronized version of GSP as follows and term it GSP2.

Algorithm 2 (GSP2):

- Each node independently generates a random time interval and chooses either going to sleep with probability p or staying awake with probability $1-p$ for the interval.
- Every sleeping node wakes up at the end of its own interval
- Every node repeats the above process for every random interval independently

Unlike other protocols using sleep mode (e.g., cluster-based schemes, SPAN and GAF), GSP is extremely simple and requires almost no information, even from immediate neighbors. The gossip sleep probability p is purely dependent on the network density and can be configured before the deployment of the network. GSP improves upon the energy consumption by schemes such as Span and GAF by not requiring nodes to transmit and receive additional network maintenance traffic. On the other hand, GSP is expected to provide less improvement on the network lifetime than other schemes due to the limited knowledge of the network, which contributes to the simplicity as we just mentioned. Therefore, GSP is more suitable to the large low-cost network, which desires low complexity to reduce the cost of every node as much as possible.

The major objective of GSP is to achieve energy efficiency by putting some nodes in a sleep mode. The potential disadvantage of this approach is that packets may go through longer paths if the nodes sleeping are on the shortest paths between source and destination nodes, resulting in more energy consumption in the network-wide communication. Also, paths will be broken more often due to mode change of the nodes. Therefore, more overhead is generated to overcome the path failures and this will consume some extra energy. So we are concerned if the energy saved by GSP is larger than the extra energy consumed by non-optimal paths and extra routing overhead. In addition, path failures due to sleeping will decrease the network throughput and increase the end-to-end delay. In order to evaluate the tradeoffs with GSP, we have conducted a discrete event simulation based performance study.

IV. PERFORMANCE EVALUATION

If we assume that there are no traffic and routing overhead in the network, all awake nodes stay in the idle mode and the sleep mode does not consume any energy, then with gossip sleep probability p the network lifetime should ideally be prolonged by a percentage of p. In practice, the improvement will be less than p. One reason is that the sleep mode consumes non-zero energy although it's very small compared with idle mode. More importantly, the longer paths caused by non-optimal routing and the extra routing overhead caused by more frequent path failures will consume extra energy compared with routing without GSP. The value of p depends largely on the density of a network. Therefore, we expect to see a larger improvement in a denser network. Additionally, since the GSP is able to maintain the connectivity of the network with a proper value of p, the throughput and packet end-to-end delay are not expected to be affected too much given a light or moderate traffic load. To confirm the above analysis, we use simulation to study the performance of GSP in the remainder of this section.

TABLE I

ENERGY CONSUMPTION MODEL FOR LUCENT IEEE 802.11 WAVELAN PC CARD WITH $2Mbps$

Radio mode	Energy Consumption (W)
Transmit	1.327
Receive	0.967
Idle	0.844
Sleep	0.066

A. Simulation model

We utilized the ns-2 network simulator [20], with CMU Monarch Project wireless and mobile ns-2 extensions, to study the characteristics of GSP. The distributed coordination function (DCF) of IEEE 802.11(b) for wireless LANs is used as the MAC layer. The radio model is similar to Lucent's WaveLAN, which is a shared media radio with a nominal bit rate of 2Mb/sec and a nominal radio range of 250 meters.

Note that GSP requires no information from the routing algorithms and can be integrated with a number of routing protocols. Here, we use Dynamic Source Routing (DSR) [6], [7] as an example to describe how GSP works. We attached GSP to DSR to get GSP+DSR.

The simulation results presented in this paper are based on scenarios randomly generated by CMU ns-2 extensions. We use 50 and 100 transit nodes to study the density effects and nodes are randomly placed within a $1500m \times 300m$ area. The node mobility model is Random Waypoint [7], in which each node begins at a random position, picks a new random position to which to move, and moves there in a straight line at a random speed. Each node independently repeats this behavior and the average degree of mobility is varied by making each node remain stationary for a period called pause time every time before it moves to the next position. The smaller the pause time, the higher the average mobility. In our simulation, the maximum speed of the nodes is 20 m/s and the pause time is varied between 0 and 1000 seconds. In case of 1000 seconds, the network is static. Besides the transit nodes, there are 10 traffic nodes, which are the source and the sink of the traffic. CBR traffic based on UDP is used. Each packet carries 512 bytes of data payload, making the packet size 532 bytes including an IP header. The packet rate is 10 packets/sec.

Our energy consumption model is based on Feeney and Nilsson's measurements of an IEEE 802.11b Lucent WaveLAN wireless network interface operating in an ad hoc networking environment [21]. Their measurements are summarized in Table I, where we can see sleep mode costs only a tiny fraction of the costs of other modes. Other measurements in the literature evaluating other 802.11b vendor equipments show the similar costs.

To make sure the traffic does not stop before the network

TABLE II

SIMULATION PARAMETERS

Parameters	Values	Parameters	Values
Simulation time	$\geq 400sec$	Bandwidth	$2Mb/s$
Physical layer	IEEE 802.11b	Max. speed	$20m/s$
MAC layer	IEEE 802.11b	Radio range	$250m$
Traffic model	CBR	Gossip period	$20sec$
Packet size	$512bytes$	Traffic nodes	10
Network size	$50/100nodes$	Packet rate	$10pkt/sec$
Area size	$1500m \times 300m$	Pause time	$0 - 1000s$

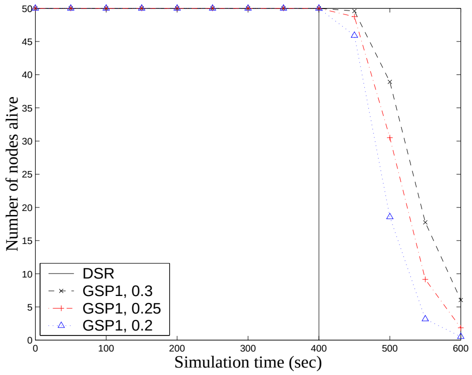


Fig. 3. Network lifetime of DSR and GSP1 with different gossip sleep probability (p)

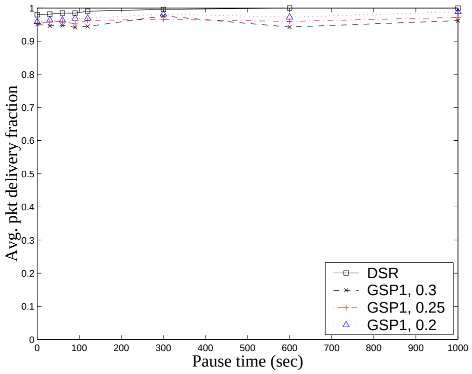


Fig. 4. Average packet delivery fraction of DSR and GSP1 versus nodal pause time

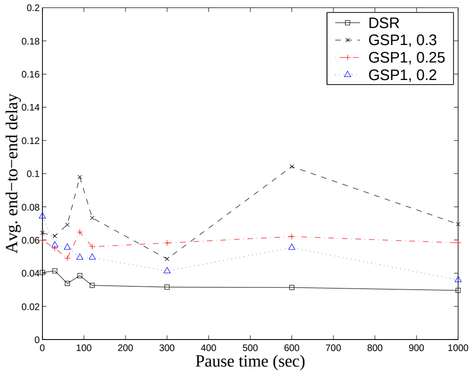


Fig. 5. Average end-to-end delay of DSR and GSP1 versus nodal pause time

dies, we give traffic nodes infinite energy. The transit nodes have enough energy so that the DSR protocol can run for 400 seconds. Since all nodes in DSR keep listening even without traffic, they run out energy almost at the same time. Also, to mitigate the effects from traffic nodes, we make traffic nodes neither run GSP nor forward traffic in DSR. However, traffic nodes do follow the same mobility model as transit nodes and maintain their own connections as required by DSR.

The parameters for GSP are chosen to show the properties of GSP and they are not necessarily the optimal values. We assume synchronization for GSP1 and use a fixed 20 seconds as the gossip period. The gossip sleep probability is varied as shown in the figures. Each data point is an average of at least ten runs with different sets of initial node placements and random waypoints. The above simulation parameters are summarized in Table II.

B. Simulation results

We evaluated three performance metrics defined as follows:

- Packet delivery fraction: ratio of the packets delivered to the destination to those generated by the CBR sources;
- End-to-end delay: Delay experienced by each packet, including queuing delays, route discovery delays, retransmission delays at the MAC layer and the salvage process of DSR, and propagation delays;
- Network lifetime: the simulation time that a fraction of transit nodes are alive and the network maintains an acceptable packet delivery fraction and end-to-end delay.

To illustrate the basic idea of GSP, we first compare GSP1 with DSR in a network of 50 transit nodes. Fig. 3 shows the number of nodes alive with respect to the simulation time. GSP1 successfully extends the network lifetime and larger sleep probabilities generate longer extensions. It is worth noting that GSP is independent of the pause time. Although not presented here, the simulation results show that, given a sleep probability, the results for different pause times are almost same. So every curve in Fig. 3 is an average of all results for multiple levels of mobility, from static to constant moving.

Fig. 4 and Fig. 5 show the packet delivery fraction and end-to-end delay with respect to the pause time for the first 400 seconds. After this time DSR fails totally and comparing the entire time would be unfair for DSR. As we expected, with a small gossip sleep probability, GSP does not significantly affect the performance of DSR in terms of average packet delivery fraction. In terms of mean end-to-end delay as expected the delay increases with p as compared to DSR.

To study the effects of network density, we run the similar simulation in the network of 100 transit nodes with pause time 0. With a higher density, a larger sleep probability can be employed and the network connectivity can still be maintained. Again, Fig. 6 shows that the lifetime extension is proportional to the gossip sleep probability and GSP can benefit from the network density. Fig. 7 shows our results of monitoring the packet delivery fraction for every 50 seconds for GSP in the same scenario. We can see, although the average

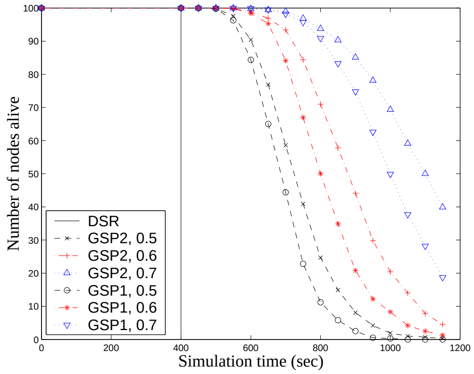


Fig. 6. Network lifetime of DSR, GSP1 and GSP2 with different gossip sleep probability (p) in a network of 100 transit nodes with pause time 0

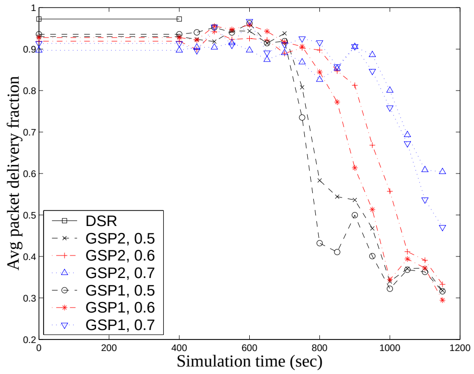


Fig. 7. Average packet delivery fraction of DSR, GSP1 and GSP2 with different gossip sleep probability (p) in the network of 100 transit nodes with pause time 0

packet delivery fraction of GSP is slightly decreased, it can be maintained for a much longer time.

In both Fig. 6 and Fig. 7, we also show the results of GSP2. The gossip interval in GSP2 is uniformly distributed between 0 and 40 seconds, with an average of 20 seconds. The results show that GSP can work well even without synchronization. Actually, the figures show that GSP2 achieved slightly better performance than GSP1. We attribute this to the fact that higher randomness can smooth and more evenly distribute the power consumption of a network.

V. CONCLUSIONS AND FUTURE WORK

This paper has proposed a novel sleep management approach for both sensor and mobile ad hoc networks to reduce energy consumption. We achieved the simplicity by only adding a timer to each node. Once its time is up, every node decides whether to go to sleep in the next period with the gossip sleep probability p. The property of gossiping makes it scalable to very large networks, theoretically, even to a network with infinite number of nodes. Network connectivity is decided by the gossip sleep probability p. We can see that certain values of p make almost all the awake nodes in the network connected. This is also shown by the simulation results. On the other hand, the performance of the network is only slightly affected. Another advantage of GSP is that the energy consumption is more evenly distributed in the entire network since the nodes go to sleep in a fully random fashion and the traffic forwarding continuously via the same path can be avoided. Further work is required to address various properties of GSP, e.g. the value of the gossip sleep probability to avoid network partition in different types of topologies and the tradeoff of energy savings versus the increased network delay.

REFERENCES

[1] C. E. Perkins and P. Bhagwat, “Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers,” in *Proceeding Of ACM SIGCOMM*, Oct. 1994, pp. 234–244.
[2] S. Murthy and J. J. Garcia-Luna-Aceves, “An efficient routing protocol for wireless networks,” *ACM Balzer Mobile Networks and Applications Journal*, 1996.
[3] T. Clausen, P. Jacquet, J. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized link state routing protocol,” *IETF Internet Draft*.
[4] B. A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T. W. Chen, “Scalable routing strategies for ad hoc wireless networks,” *IEEE Journal on Selected Areas of Communications*, vol. 17, no. 8, Aug. 1999.
[5] E. M. Royer and C.-K. Toh, “A review of current routing protocols for ad hoc mobile wireless networks,” *IEEE Personal Communications*, Apr. 1999.
[6] D. Johnson, “Routing in ad hoc networks of mobile hosts,” in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1995.
[7] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” 1996.
[8] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
[9] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *Proceeding of IEEE INFOCOM*, Apr. 1997.
[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *IEEE Hawaii International Conference on System Sciences*, 2000.
[11] A. Manjeshwar and D. P. Agrawal, “TEEN: A routing protocol for enhanced efficiency in wireless sensor networks,” in *IEEE International Parallel Distributed Processing Symposium*, 2001.
[12] A. Manjeshwar and D. P. Agrawal, “APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks,” in *IEEE International Parallel Distributed Processing Symposium*, 2002.
[13] 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*, no. 8, pp. 481–494, 2002.
[14] Y. Xu, J. Heildemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proceeding Of ACM SIGMOBILE*, 2001.
[15] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Topology management for sensor networks: Exploiting latency and density,” in *Proceeding Of ACM MOBIHOC*, 2002.
[16] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-based ad hoc routing,” in *Proceeding of IEEE INFOCOM*, 2002.
[17] R. Meester and R. Roy, *Continuum Percolation*. Cambridge University Press, 1996.
[18] G. Grimmett, *Percolation*. Springer-Verlag, 1989.
[19] F. Ye, A. Chen, S. Lu, and L. Zhang, “A scalable solution to minimum cost forwarding in large sensor networks,” in *Proceedings of the International Conference on Computer Communications and Networks*, 2001.
[20] K. Fall and K. Varadhan, *The ns Manual*. http://www-mash.cs.berkeley.edu/ns/, 2002.
[21] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *Proceeding of IEEE INFOCOM*, 2001.