Floossiping: A New Routing Protocol for Wireless Sensor Networks

Yuecheng Zhang and Liang Cheng
Laboratory Of Networking Group (LONGLAB)
Department of Computer Science and Engineering, Lehigh University
Bethlehem, PA 18015, USA
{zhy2@lehigh.edu, cheng@cse.lehigh.edu}
http://long.cse.lehigh.edu/

Abstract-***In this paper we present a new routing protocol, floossiping, for wireless sensor networks. It can be regarded as an enhancement to existing flooding and gossiping approach by using a single branch gossiping with low-probability random selective relaying in order to achieve a better overall performance. By letting each sensor node decide its own activity in the routing procedure, we implement a simple zero-overhead resource-aware routing protocol. It has the advantages of flexibility in delay-power tradeoff, affordability in terms of resource consumptions, and reliability in terms of packet lost-free. Simulation results show that a range of optimal operating points considering delay and power consumption tradeoff can be found with user-customized cost functions.***

Key words: floossiping, flooding, gossip, delay, power, routing, wireless sensor networks

I. INTRODUCTION

A sensor network is a self-organizing cooperative network, which is composed of a large number of sensor nodes that are densely and sometime randomly deployed either inside the phenomenon or very close to it, which is monitored by a manager node [1]. Each sensor node is capable of sensing phenomena, collecting and processing data, as well as reporting/routing the data back to a data sink. The data sink relays the data to the manager node via inter-networks.

Conventional network protocols, when applied to wireless sensor networks, face challenges of low complexity and high reliability requirements as well as resource-saving constraints. In this paper, we propose an original routing protocol called *floossiping* for sensor networks, which has the advantages of flexibility in delay-power tradeoff, affordability in terms of resource consumptions, robustness for dynamic network topologies, and reliability in terms of packet lost-free.

The paper is organized as follows. In section 2, we describe the goal and motivation of developing a new network protocol. Section 3 classifies and briefs various approaches for wireless sensor networking. In section 4, we propose a simple, highly reliable yet flexible routing protocol, called *floossiping*. The network establishment, routing process as well as failure-handling issues are carefully addressed followed by the verification of reliability. Preliminary simulation results and some discussions on the performance of *floossiping* are shown in section 5. We conclude the paper in section 6 with a brief list of further research topics.

II. EXISTING APPROACHES

There are quit a few existing routing protocols have been proposed for wireless sensor networks.

A. Protocols based on Local Information

Flooding [2]: In flooding, on receiving a data or management packet, a node will repeat it by broadcasting unless the node is the destination of the packet or the Time-To-Live (TTL) has been reduced to zero. It is one of the most traditional routing schemes, and due to its simple implementation, practical nature and the guaranteed minimum delay, it is widely used as a basic routing protocol or as an important part of more complex protocols. However, flooding does need a fine tune of TTL to handle the traffic implosion, overlapping, and source blindness. With limited topology information, the tuning objective is not always achievable. Moreover, besides one-to-all services, the flooding is not efficient enough for one-to-one message delivery.

Gossiping [7]: To reduce the power consumption while keep the routing system as simple as possible, gossiping was proposed as an alternative to flooding. In gossiping, only a selected subset of neighbors of a node will repeat a packet that they have received from the node. Since the total power consumed can be approximate as $O(k^L)$, where k is the number of neighbors repeat the packet and L represents the number of hops before the relaying process stops. The most desirable feature of gossiping is that power consumption can be controlled by selecting appropriate k and L. Meanwhile, delay of gossiping routing is relatively longer than the optimal value and sometimes too long for practical applications. In contrary to flooding, gossiping is good at one-to-one communication scenarios but generally it does not serve well for one-to-many scenarios.

Sensor Protocols for Information Via Negotiation [9]: The node with data to send will broadcast a much shorter advertisement packet (ADV) with a descriptor for the data and then wait for request packets (REQ) from its neighbors, then send data packets (DATA) to all nodes that have sent corresponding requests. While this is a data centric protocol, we notice that its performance deteriorate rapidly with the increase of event possibility, i.e. while more data need to be transmitted, the traffic overhead increase explosively. The efficiency will be extremely low especially in general many to one cases, such as many sensor nodes report data to a single data sink in sensor networks.

B. Tree-based Routing Protocols

Clustering Hierarchy Protocols: In clustering hierarchy protocols, some sensor nodes are selected to be the local cluster-heads that take charge of collecting and relaying information from subsets of local nodes to the sink or upper-level cluster-heads. By applying hierarchical structure, this sub-class simplifies some otherwise complicated jobs such as channel reuse and synchronization. However, a better scheme for cluster-head polling and managing, which can take full advantages of its potential strength, is still under study.

Small Minimum Energy Communication Network [10]: The basic idea is to build an energy-efficient sub-graph of the graph composed of all sensor nodes and the sub-graph is taken as the routing set. While it is very energy-efficient, to find the optimal sub-graph is unfortunately a NP-hard problem.

C. Source Routing Protocols

Sequential Assignment Routing (SAR) [11]: In SAR, each node selects its own next hop to the sink according to QoS metrics, power resources, and packet priority. Although it is quite straightforward in concept, its source routing nature makes it necessary to distribute and store non-trivial amount of global information at each node, which may in turn increase the overhead of the network as well as the cost of each node.

Directed Diffusion [12]: directed diffusion is based on following ideas. First, the sink broadcasts its interests all over the network. Along with the propagation of interest a gradient field covering the whole sensor area is built which later serves as a routing guidance. Although every source only knows the optimal next-hop or next-several-hops information instead of a complete route to the sink, the route is determined by each node locally and thus belongs to source routing category and share the same problem as SAR.

III. MOTIVATION

The goal of this research is to develop a routing protocol that is both affordable in implementation and flexible in QoS provision. The protocol will support densely deployed network with dynamic architecture. Major factors that we have considered in designing the new protocol include: simplicity in implementation, reliability under the condition of high single-node failure ratio, resource awareness, and flexibility to fit in various delay and power-consumption constraints.

Simplicity: We aim to provide a lightweight routing protocol that can help to provide a simple solution for routing packets within wireless sensor networks. It can serve as either an aid of any more complicated yet more efficient routing protocols by delivering the routing maintenance packets or as an affordable routing solution for applications in scenarios of highly resource-constrained sensor networks. Thus, we try to keep the protocol as simple as possible by trading off complicated functionality. As illustrated in later sections, the simplicity consideration does not refrain us from providing a balanced controllable combination of the flooding and gossiping protocols, which is capable of serving both one-to-one and one-to-many communication scenarios yet keeping the number of working states at s sensor node as few as possible.

Reliability [5]: One of the major advantages of a wireless sensor network is its easiness in deployment because (*i*) there is no sophisticated wiring; (*ii*) the system is more robust under single node failure, if enough redundancy is kept; and (*iii*) replacing a malfunctioned or out-of-power sensor node becomes easier. However, all these benefits require powerful and well-defined network establishment and maintenance functionality. As we all know, the working environment of wireless sensor networks is harsher and unpredictable compared to traditional wireless networks. Therefore, how to provide a reliable routing service becomes one of the most important and challenging tasks of our design. Mechanism is implemented to guarantee the property of packet-loss free under undesirable working conditions.

Resource Awareness [6][7]: To make it even desirable, routing and operating mode decision at a sensor node should always be made with the knowledge of resource status of sensor nodes within certain limited areas. Thus, in the proposed protocol, we provide several parameters that can be used to control the decision making process. Although we do not explore individual resource control algorithm in this paper as it is not the focus of this paper, we verify in section IV that various resource-aware intelligence can be easily integrated into the proposed protocol by control these parameters.

Flexibility: Actually, those control parameters can provide more than just capability of integrating resource-aware intelligence into the wireless sensor networks. Thus they give us more room for tradeoffs between different QoS provisions, such as routing delay and power consumption. Different networks or different nodes within a network can adjust parameters according to their knowledge about the application expectation, the network deployment, the working environment, the resource condition of individual node and so on.

IV. FLOOSSIPING

A. *Basic Idea*

The basic ideas of *floossiping* are (*i*) combining a single branch gossiping and a controllable low-probability random selective relaying (LPRSR) to achieve a better overall performance and (*ii*) letting each sensor node decide its own activity in the routing procedure to achieve a zero-overhead resource-aware routing. The LPRSR branch is controlled by TTL such that the overall power consumption can be constrained. The gossiping branch is not controlled by TTL and the gossiping process will not stop until the packet reaches its destination, which guarantees the reliable packet delivery. Note that it is possible for duplicate packets arrive at the destination node. Thus we assume that it is the upper layer's responsibility to distinguish the duplicated packets and properly handle the duplication by reading the information released by the protocol within the payload, namely, source address and sequence number. The objective is to keep the simplicity of the protocol and to reduce the buffer consumption.

There is an entry at each node for the address of its next hop neighbor denoted as A_{RSN} and an entry for the node's own address denoted as A_{OWN}.

A packet will have a 8-tuple (A_{SRC}, A_{dest}, SN, TTL, TH_{RSR}, A_{Next}, A_{Last}, MODE) in its header. The tuple represents source address, destination address, packet sequence number generated by its source, time-to-live of the packet, threshold assigned by the source to control the LPRSR procedure, nominated next hop address, current sender's address, and a Boolean mode to differentiate the gossip branch and LPRSR branches.

When MODE equals to 1, it means that the packet is traveling on a gossip branch and the relay will never stop until the packet reaches its destination and the destination sends out a specific empty packet to confirm the receiving of the packet. When MODE equals to 0, it implies that the packet is on a LPRSR branch and the transmission will under control of TTL value.

The random selective relay threshold TH_{RSR} is set by the source of the packet to determine how the packet is supposed to be delivered. With a larger TH_{RSR}, more neighbors will join the random selective relay process and in turn reduce the expected minimum number of hops. While TH_{RSR}=0, the proposed protocol works exactly like a single branch gossiping without TTL. While TH_{RSR}=1, the protocol will work as flooding if the TTL value is large enough. Actually the gossiping branch will not stop transmitting when TTL expires. However, we regard it as the cost to compensate the possible reliability problem due to an overly small TTL value.

When a new node is added into the system, its next hop neighbor address will be void until the node receives a data packet. It will initiate the entry into A_{Last}. And then it will get into ready state to serve as a router in the system.

B. *Finite State Machine*

The finite state machine of the floossiping is depicted as Figure 1. There are two working states, namely, "ready" and "waiting repeat". While a sensor node receives a packet with MODE=1 and finds out that it is not the destination of the packet, it must relay the packet to its next hop neighbor and move from "ready" state into "waiting repeat" state. In "waiting repeat" state, the node will passively listen to the network to determine whether the gossiping packet is properly handled by the next hop neighbor. Whenever a valid relay packet is heard, the state will resolve and the node will come back to its "ready" state.

While a node is in the "ready" state, on receiving a packet, the node will do the following steps in order:

1. Compare the A_{OWN} with the A_{dest}. If they match, then accept and release the data to the upper layer. And if MODE=1 then it sends an empty packet (A_{SRC}, A_{dest}, SN, TTL-1, TH_{RSR}, A_{dest}, A_{dest}, 1)); otherwise,
2. If MODE=0, jump to step 4.
3. Compare A_{OWN} with A_{Next}. If they match, repeat the data packet with a header as (A_{SRC}, A_{dest}, SN, TTL-1, TH_{RSR}, A_{RSN}, A_{OWN}, 1) and jump to step 6.
4. If TTL<=1 then jump to step 6.
5. Generate a random number, multiply a weight, and denote the result as $WR_{\mathrm{Re}lay}$. Compare the $WR_{\mathrm{Re}lay}$ with TH_{RSR}. If $WR_{\mathrm{Re}lay} > TH_{RSR}$, repeat the data packet with a header as (A_{SRC}, A_{dest}, SN, TTL-1, TH_{RSR}, A_{RSN}, A_{OWN}, 0). In this step, we have two parameters to control. One is the random selective relay threshold TH_{RSR}. The other is the weighted random number $WR_{\mathrm{Re}lay} = W \times Random$, in which *W* is a weight decided by the sensor node according to its resource conditions such as power, computing capability, and transmission capacity. By taking different values of *W*, applications can customize the routing protocol to meet their own resource interests.
6. Refresh the A_{RSN} in a manner that if either a random number is larger than the threshold *TH*

and MODE=0 or the $A_{Next} = A_{Last}$ then $A_{RSN} = A_{Last}$. *TH* is used to control the refreshing of A_{RSN}. A larger *TH* means that A_{RSN} will keep consistent for a longer period of time and a smaller *TH* makes A_{RSN} refresh more frequently. By properly selecting the value of *TH*, trade-off between fairness in job distribution and QoS consistency can be achieved.

After sending a packet (A_{SRC}, A_{dest}, SN, TTL, TH_{RSR}, A_{Next}, A_{OWN}, MODE), if MODE=1 then the sender will expect to hear the repeat packet (A_{SRC}, A_{dest}, SN, TTL-1, TH_{RSR}, *, A_{Next}, 1) for a certain period of time. If the timer expires, the sender will move into a non-available state (set A_{RSN} as VOID), and begin to listen to the network for next available data packet (A_{SRC}', A_{dest}', SN', TTL', TH_{RSR}', A_{Next}', A_{Last}', MODE'). Then set $A_{RSN} = A_{Last}$' and send the packet (A_{SRC}, A_{dest}, SN, TTL, TH_{RSR}, A_{Last}', A_{OWN}, 1). If MODE=0, then the sender immediately gets ready for the next transmission for data relay.

C. Reliability

If a node X is temporarily unavailable for some reasons, there are four possible resulting scenarios:

1) No node has A_{RSN}=X, then there won't be any problem.

2) Some nodes has A_{RSN}=X, and all these nodes still have link(s) to the sink, then retransmissions will solve the problem and data won't get lost.

3) Some nodes has A_{RSN}=X, and some of these nodes, say Y, is isolated from sink(s) due to the failure of X; but during the period of failure, no packet is in Y waiting to be send or relayed. There is no data to be lost in this scenario.

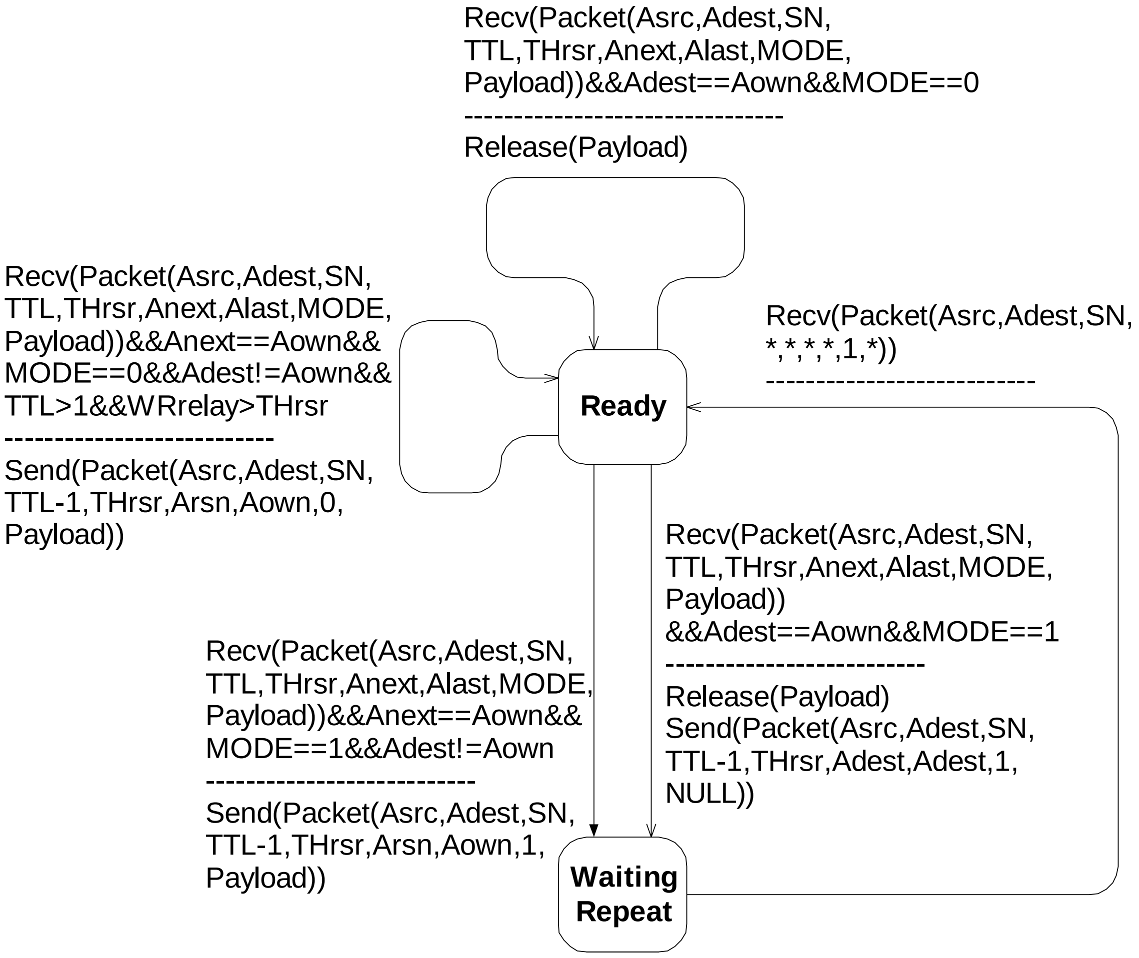

Figure 1. Finite state machine of Floossipping

4) Some nodes has A_{RSN} =X, and some of these nodes, say Y, is isolated from sink(s) due to the failure of X; but during the period of failure, there are some packets in Y waiting to be send or relayed. Then since Y is separated from the sink, the single branch gossiping will either continue going within Y or be blocked within one non-available node. The packet(s) will never get lost, and once the failure is resolved by either X comes to live again or the introduction of some new nodes, the packet(s) will be sent right after the reconnection of Y to the sink(s).

D. Working Procedure

Figure 2 depicts three continuous hops during a floossiping routing process. Four parameters (TTL, A_{Next} , A_{Last} , MODE) are shown in each transmission. The first step begins by node#1 broadcasting a gossip packet to its next hop neighbor node#3. Node#3 gets this broadcasting and continues the gossiping branch by broadcasting a gossip packet to its next hop neighbor node#10. At the same time, all neighbors of node#1 except node#3, say, node#2, node#4, node#5, node#6, and node#7, also receive node#1's broadcasting packet and begin the LPRSR relaying process. Some of them (e.g., node#5 and node#7) will decide to relay the packet by broadcasting a relaying packet with MODE=0 and TTL-1. The next step includes that node#10 continues the gossiping broadcast and node#4, node#11, and node#12 continue the LPRSR branches. It is noticed that node#5 receives the broadcasting packet again from node#4 during the third round of the transmission. Therefore it will transmit a LPRSR packet in the fourth round of the transmission again, thus a duplicated packet is generated.

V. SIMULATION RESULTS

A. Assumptions

In the following simulation, we assume that (*i*) the transmission is always done in the simplest way, i.e. by omni-directional broadcast; and (*ii*) there is no power control capability at the transmitter, i.e., all nodes will have a round coverage range with a uniform radius. Thus the power consumption of the data transmission is approximately proportional to the number of total bits transmitted, including payload as well as protocol headers.

The size of data packets transmitted within the wireless sensor network is assumed to be the same and the data payload has a larger size than the 16-byte header. Therefore, the transmitting, receiving, and processing power consumptions are approximately proportional to the total number of packets transmitted. Furthermore, consider that the power consumption is consisted of the power consumed by transmitting, receiving and processing packets and a roughly constant power consumption that is generally one or two magnitudes smaller than the power consumption of transmitting/receiving and processing packets. As a result, we assume that the power consumption of the protocol is approximately proportional to the total number of packets transmitted.

Also we assume that there is a perfect error-free MAC layer. Thus when considering delay, it comes from our protocol only and is approximately proportional to the minimum number of hops among all branches.

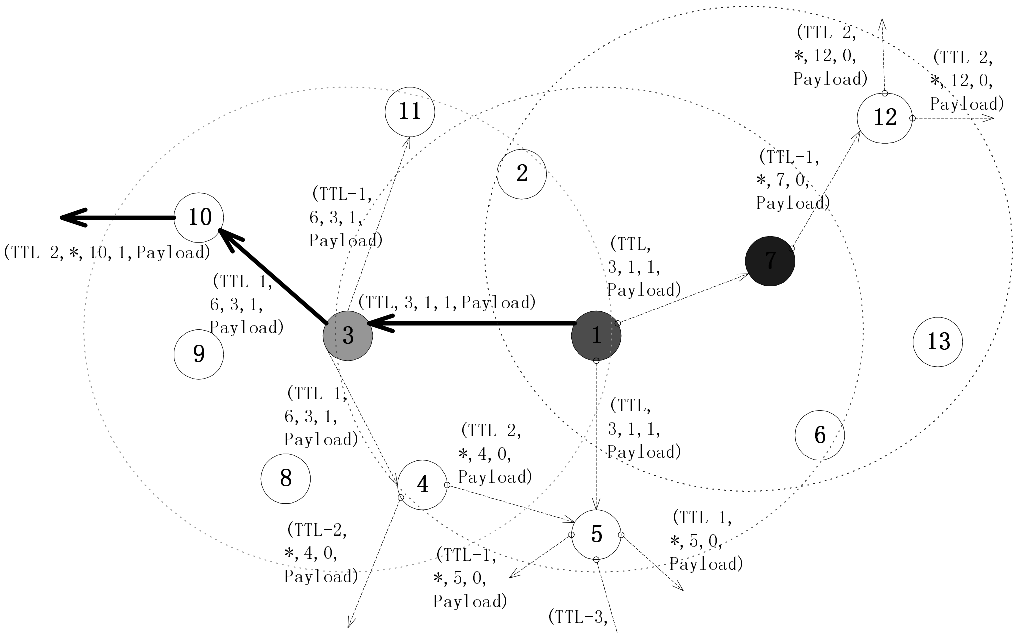

Figure 2. Working procedure of floossiping

B. *Simulation Configuration*

Monte Carlo simulations of the protocol are performed with following steps:

1) Within a $a \times a$ virtual area, we randomly select N positions to place sensor nodes.

2) Assume the covered area of each sensor nodes is a circle of radius r, say, any nodes i and j that satisfying $d_{euler}(i, j) \le r$ will be in each other's neighborhood. Assume that the network topology is static and the network establishment procedure has been finished. Also assume that each node will work in a perfect status, i.e. weight for computing $WR_{\mathrm{Re}lay}$ is always 1.

3) Sensor 0 will send k independent packets to sensor $N-1$. Since sensors are randomly deployed, the selection of source and sink won't harm the generosity of the simulation. We count and average the number of hops for each packet to reach the sink for the first time and the total number of transmissions happened for each packet.

To simplify the process, the refresh of the next hop address is supposed to be done in pure random way. In other words, every time a node will select a random next hop node from all its neighbors. By this way, we can focus on the traffic between node 0 and $N-1$ while eliminate all other traffics, which are necessary in a real implementation.

C. *Simulation Results*

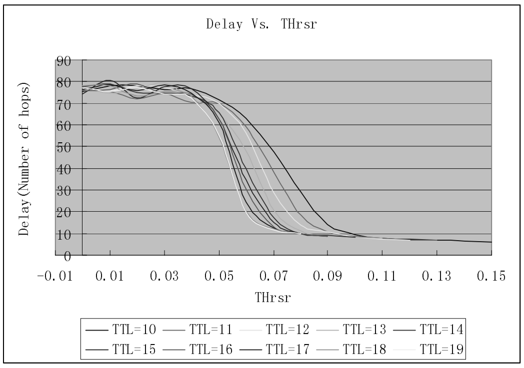

Figure 3. Delay Performance of *floossiping* (N=100, a=10, r=3, K=1000, TTL=10-19)

Figure 3. shows the delay performance as functions of TH_{RSR} under certain parameters for various TTL values. It is obvious that the delay begins to drop fast as TH_{RSR} go beyond certain values, which is determined mostly by the network deployment. Then the delay value keeps decreasing until it converges to the optimal value. These curves show that (*i*) Adding the LPRSR branches will improve the delay performance of gossiping even with a relatively small TH_{RSR}. (*ii*) With the same network topology, the larger TTL value is the faster the curves converge to the optimal value. This is reasonable because larger TTL implies deeper LPRSR branches which in turn result in a large probability for a packet to reach the destination node if the destination node is reachable within TTL number of hops(i.e. TTL>Minimum number of hops from source to destination). (*iii*) The flat area begin from zero is due to the limited number of neighbors for any given nodes, while the product of TH_{RSR} and number of neighbors is small enough, almost no LPRSR branch will grow up during the routing process and thus the gossiping branch will be in dominant in calculating the delay. (*iv*) With the increase of TH_{RSR}, more and more LPRSR branches appear and they become less and less negligible in determining the delay, which result in the fast drop area. (*v*) With the number of LPRSR branches increasing further, probability that at least one of the LPRSR branches will pass the destination nodes increase fast and the curve will converge to the minimum number of hops decided by the network deployment.

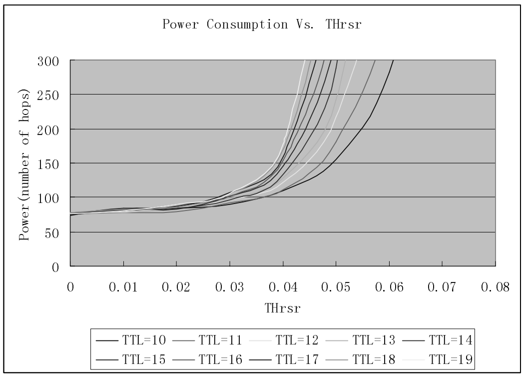

Figure 4. Power Performance of *floossiping* (N=100, a=10, r=3, K=1000, TTL=10-19)

Figure 4. shows the power consumption as functions of TH_{RSR} under certain parameters for various TTL values. It demonstrates that (*i*) the increase of power consumption is quite slow at the beginning, and it keeps accelerating afterwards. This implies that the introduction of a LPRSR into a gossiping protocol is acceptable at low probability range as we claimed. Thus, as a complement to gossiping, RSR should always be limited to low probability range because the gain comes with increasing TH_{RSR} will become trivial when the probability is in a high range, while the power punishment keeps increasing in an exponential manner. (*ii*) The smaller TTL is the faster the overall power consumption will increase.

To further illustrate the search of optimal threshold TH_{RSR} while considering both delay and power matrices, we need to select a cost function that consider both delay and power factors. To avoid the non-uniform unit matching problem, we begin with taking the product of delay and power (number of transmission and minimum number of hops to reach the destination node in our case and it should be a good estimation as we explained in section 5A) as the cost function for our optimizing. Figure 5. shows the product as functions of TH_{RSR} under certain parameters for various TTL values. It can be seen that: (*i*) for curves with too large(TTL>16) TTL, the curves are monotonic and the optimal value for delay power product. This is quite intuitive, since the explosive increase of power can easily beat the decrease of delay in the product with over deep LPRSR branches. (*ii*) For just sufficient (10<TTL<14) TTL, the product curves are convex and provide a single optimal TH_{RSR} . (*iii*) For some other TTL values in middle range (14-16), the curves are more complex. And where the optimal point exist is hard to predict.

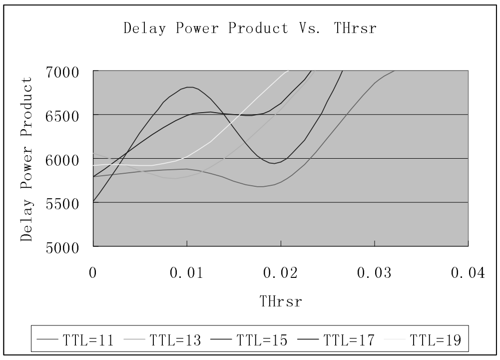

Figure 5. Delay power product and TH_{RSR} relation of *floossiping* (N=100, a=10, r=3, K=1000, TTL=10-19)

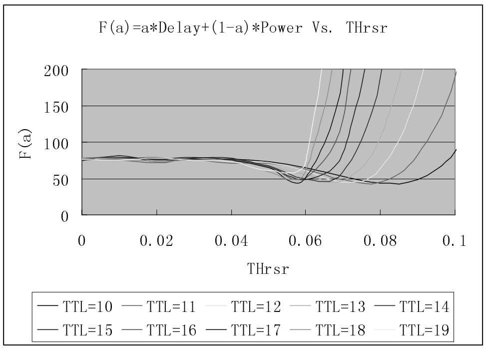

Figure 6. Linear cost functions Vs. TH_{RSR} relation of *floossiping* (N=100, a=10, r=3, K=1000, a=0.9)

Actually, the delay power product is only one of the possible cost functions can be used in optimization point search, in Figure 6. a weighted linear combination of the two measures $F(\alpha) = \alpha \times D + (1-\alpha) \times P$ is depicted with different TTL. A series of convex curve is showed and the larger TTL is the smaller the optimal TH_{RSR} under this cost function will be. It shows how different cost function can be considered during the optimization of floossiping.

VI. CONCLUSION

In this paper we present a new routing protocol, *floossiping*, for wireless sensor networks. It combines a single branch gossiping and a controllable low-probability random selective relaying (LPRSR) to achieve a better overall performance and let each sensor node decide its own activity in the routing procedure to achieve a zero-overhead resource-aware routing. Its salient features include: suitable for dynamic topology, reliable for data transmission, affordable and simple, and efficient in delay-power tradeoff. Simulation results show that a range of probabilities of LPRSR could always be found that generate optimal power-delay product.

Future works include refining practical next hop refreshment mechanism, finding a good hazard resolving mechanism and implementation in a sensor network test-bed.

REFERENCE

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., ``A Survey on Sensor Networks, IEEE Communications Magazine," Vol. 40, No. 8, pp. 102-116, August 2002.

[2] S.M. Hedetniemi, S.T. Hedetniemi, and A. Liestman, “A Survey of Gossiping and Broadcasting in Communication Networks”, Networks, vol. 18, 1988.

[3] Seong-Hwan Cho and A. Chandrakasan, "Energy Efficient Protocols for Low Duty Cycle Wireless Microsensor Networks", *ICASSP 2001*, May 2001.

[4] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Routing Protocols for Wireless Microsensor Networks," *Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.

[5] Alberto Cerpa and Deborah Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," In *Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 23-27 2002.

[6] Deepak Ganesan, Ramesh Govindan, Scott Shenker and Deborah Estrin, Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks, Mobile Computing and Communications Review (MC2R) Volume 1, Number 2, 2002.

[7] B. Krishnamachari, Y. Mourtada, and S. Wicker, "The Energy-Robustness Tradeoff for Routing in Wireless Sensor Networks," *Autonomous Networks Research Group Technical Report ANRG TR 02-001*, September 2002.

[8] J. M. Rabaey et al., “PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking”, IEEE Comp. Mag., 2000, pp. 42-48.

[9] W. R. Heinzelman, J. Kulik, and H. Balakrishnan,”Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” Proc. ACM MobiCom’99,Seattle, WA, 1999, pp. 174-85.

[10] V. Rodoplu and T. H. Meng, ¡°Minimum Energy Mobile Wireless Networks,¡± IEEE JSAC, vol. 17, no. 8, Aug.1999, pp. 1333-44.

[11] K. Sohrabi et al.,”Protocols for Self-Organization of a Wireless Sensor Network,” IEEE Pers. Commun., Oct. 2000, pp. 16-27.

[12] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, to appear in Proc. ACM Mobicom, Boston MA, August 2000