Energy-Efficient Routing Protocol in Event-Driven Wireless Sensor Networks

Yan Sun, Haiqin Liu, and Min Sik Kim
School of Electrical Engineering and Computer Science
Washington State University
Pullman, Washington 99164-2752, U.S.A.
Email: {ysun,hliu,msk}@eecs.wsu.edu

***Abstract*—Routing is critical in wireless sensor networks (WSNs). It has been widely studied in recent years and many new protocols have been proposed. In this paper, based on observations on event-driven wireless sensor networks, we propose algorithms to reduce power consumption and improve data quality in wireless sensor networks. In our algorithms, sensor nodes reduce the sampling frequency and number of hops when there is no event, and the positive feed-back scheme wakes up sensor nodes quickly to capture an event. Our algorithms add information in packets and use negative-ACK packets instead of ACK packets to reduce bandwidth consumption.**

I. Introduction

Wireless sensor networks (WSNs) become more and more popular and have been widely used recently [1]. WSNs usually consist of a large number of sensors to accomplish sensing tasks for military, healthcare, civil, environmental, and commercial applications [2]. WSNs are widely used in the remote tough areas to help people to do surveillance; WSNs can also be used to monitor and analyze the motion of a tornado [3]; WSNs are used to monitor the temperature around a volcano; WSNs are used to monitor the wild animal activity. The sensor nodes are severely constrained by energy because most wireless sensors are powered by batteries [2]. Thus, the most important issue in WSNs is to design efficient energy-aware routing protocols [4]. Another important issue in WSNs is that all the sensors in the same WSN must share the limited channel bandwidth which often leads to congestion and packet drop, and many routing protocols have been proposed to reduce bandwidth consumption.

Each sensor node in WSNs usually consists of one or multiple sensors, an embedded processor or DSP, a lower-power radio transceiver, and a battery [5]. A single wireless sensor node senses physical phenomena in the environment, such as temperature, humidity, pressure, and so on. After sensing, it converts analog signal into digital signal using Analog-Digital converter (ADC), and then the embedded CPU, micro controller, or DSP processes the signal and transfer the data in packets to neighboring nodes. Each sensor node works as both a sender and a receiver. The main components in a sensor node are shown in Fig. 1. A sensor node can make a decision on the event and then send the information to the sink node/center fusion node/base station (We use "sink node" throughout this paper). The sink node makes the final decision based on the information received from other sensor nodes.

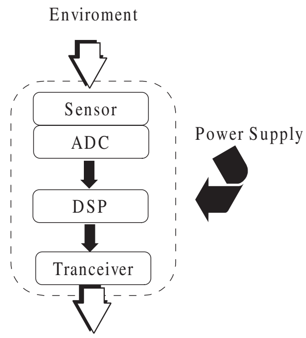

Fig. 1. The main components in a sensor node

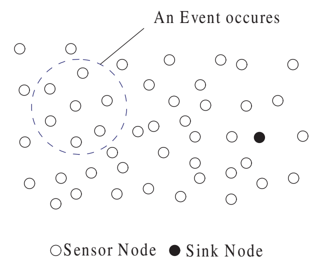

Fig. 2. A simple event-driven wireless sensor network

The sink node is usually much more powerful than sensor nodes and works as an interface with the administrator.

In event-driven wireless sensor networks, data is sent to the sink node whenever an event occurs [2], but some sensor nodes may send packets to sink node when no event occurs due to the noise in the environment, and some sensor nodes may not send packets when an event occurs. Maybe only a subset of the total sensor nodes can detect the event as shown in Fig. 2. Besides, different nodes detect different signal strength of the event.

A major difference between wireless sensor networks and wired networks is that routing in wireless sensor networks is more challenging because (i) some sensors' location can be changed, (ii) in different applications, the number of sensors, the quality of sensors, the location of sensors, and the requirements may vary, and different applications may need different protocols and routing algorithms, (iii) data collected by different sensors from the same physical phenomenon may be similar and results in redundancy between packets generated by different sensors, especially sensors in the same small area. Therefore, protocols used in traditional networks cannot be used directly in wireless sensor networks, and more specific or modified protocols are needed.

In many wireless sensor networks, real-time communication is required to react quickly after an event occurs [6]. For example, a fire-fighter needs to make a decision as soon as possible based on the information collected by the wireless sensor network. In some real-time wireless sensor network

978-1-4244-6826-3/10/$26.00 ©2010 IEEE

applications, data should be delivered to the sink node within a given time period, and the late packets will be discarded. In such situations, the end-to-end latency must be bounded and the congestion issue in the network must be solved properly. In other words, data quality is more important.

The remainder of this paper is structured as follows. Related work is presented in Section II. Section III details the design of the proposed algorithms. They are evaluated in Section IV. Finally, Section V concludes the paper.

II. Related Work

Because the power consumption and data quality are usually the most important issues in wireless sensor networks, many research efforts have been devoted to find the best tradeoff between these two issues. In [7]–[9], the authors implement an approximate querying algorithm, which mainly relies on the application specific error bound that is disseminated to each sensor node along the query path. Because much related data packets exist among sensor nodes especially within a small area, most of the time results do not need to be transmitted back to the sink node. By not sanding such packages, it is possible to reduce the frequency of packet transmission while data quality still meets the requirement.

In the event-driven wireless sensor networks, no event occurs most of the time. Therefore, how to reduce the power consumption in the idle time is another important aspect of energy saving. Some sensor nodes can go into the sleep mode when there is no event just like processes go into the sleep mode in computers and wake up later when a related event occurs. [10]–[13] belong to this kind of approaches, focusing on how to design efficient sleep/wakeup scheduling protocols. In [10], the authors designed an energy efficient S-MAC protocol, which is a modified medium-access control (MAC) protocol. The sensor nodes fall into sleep periodically and they also set the radio to the sleep mode during the transmission period of other sensor nodes. The simulation results show significant power reduction over the 802.11-like MAC. [11] proposes a similar flexible energy-efficient and low-latency protocol named DMAC, which is specifically designed for data gathering trees in WSNs. The DMAC is designed to solve the data forwarding interruption problem in earlier protocols which utilize active/sleep modes. The DMAC gives the active/sleep scheduler of a sensor node an offset, which depends on its depth in the tree to enable sensor nodes' forward packets continuously. In [12], the authors propose an Efficient Sleep Scheduling based on Application Timing (ESSAT), which exploits the timing semantics of WSNs aggressively. Their algorithms do not need to maintain TDMA schedules or communication backbones, which make the algorithm more light-weight. The simulation results show that the ESSAT can save significant energy only with a little increase in the end-to-end delay. Furthermore, their algorithms are also robust in the dynamic WSNs. A distributed, cross-layer scheduling scheme has been proposed in [13], where the schedule establishment is distributed and each sensor node utilizes its multiple layers to communicate with its parent for transmission. They design an efficient schedule for the query processing; in a processing cycle, each sensor node does sampling, computes on the sampling data, communicates, and then falls into the sleep mode. The experimental results show that their algorithm can save 50–60% of energy on the sensor nodes.

Other work focuses on how to design protocols to meet QoS requirements [14]–[16]. In [14], the Sequential Assignment Routing (SAR) algorithm has been proposed for QoS wireless sensor networks. It tries to balance between data quality and power consumption, and uses multiple paths to transfer packets instead of a single data path to avoid sensor nodes or links breakdown. The simulation results show that SAR consumes less power and achieves better fault tolerance. In [15], the authors propose BodyQoS, which focuses on reliable wireless body-area networks. BodyQoS implements an asymmetric architecture between sensor nodes and the sink node, and the scheme tries to minimize the load on the sensor nodes with limited resource. Furthermore, BodyQoS can provide adaptive resource scheduling to meet different QoS requirements. A robust sensor network has been proposed in [16] which allows the sink node to adjust the resolution of the QoS received from the sensor nodes dynamically. The main idea is to enable the sink node to transfer QoS information to every sensor node through a designated broadcast channel and then adjust to the specified optimum number of sensor nodes that should be sending information at a given time dynamically.

III. Proposed Algorithm

In our algorithms, we consider a static event-driven wireless sensor network, assuming that every sensor node in the network has a unique ID. The sensors may have different quality with different signal-to-noise ratio and different memory capability, and all of them have limited processing power and can only make a basic decision on an event. All the nodes have very limited energy and thus how to prolong the life of sensors is an important issue. We also assume that all the sensor nodes can work as senders and receivers, having bidirectional communication. Our approach is based on the following observations.

A. Observations on the WSNs

(1) In the Event-Driven Sensor Networks, events are rare in a small time period. A good example is a fire detection sensor network.

(2) In the Event-Driven Sensor Networks, when an event happens, multiple sensor nodes may be activated and send packets at the same time, which leads to data implosion and redundancy in the network [6]. Many nodes may contest for the severely limited bandwidth, which makes packet retransmissions and ACKs overload links.

(3) The delay from a sensor node to the sink node during an event-less period is different from that during eventful period because of the congestion.

(4) The sensors may have different signal-to-noise ratios, and sensors with low signal-to-noise ratio may generate more

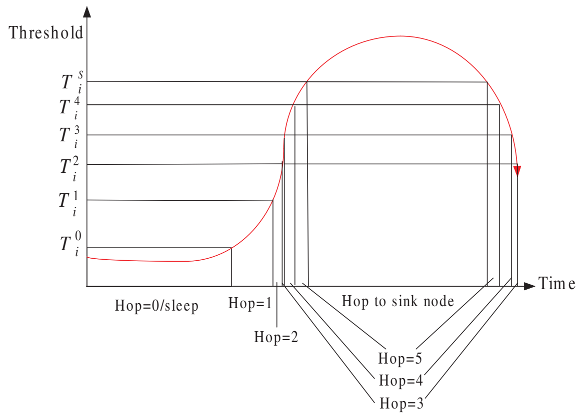

Fig. 3. The different thresholds in a sensor node

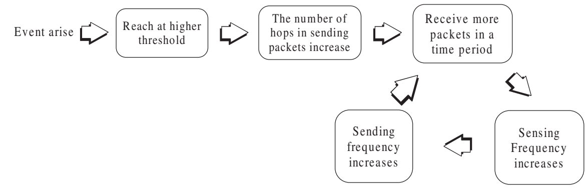

Fig. 4. Positive feed-back scheme in the wake up process

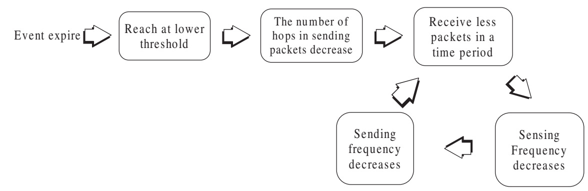

Fig. 5. Positive feed-back scheme in the fall into sleep process

fluctuating data than those with high ratio when there is no event. The latter can generate more accurate data than the former when an event occurs, which means the former should have higher priority.

(5) In the real-time WSNs, some retransmitted packets are useless because they are too late.

(6) The sink node has much more resources than ordinary sensor nodes, in terms of computation ability and memory capability. Therefore, moving tasks from senor nodes to the sink node is desirable.

B. Proposed Algorithms

Based on these observations, the proposed algorithms work as follows:

(1) Give sensors different priorities, P_0, P_1, P_2, $\ldots$, P_n, based on the quality of sensors and the positions of sensors.

(2) In order to reduce packet transmissions in the network, each sensor node i has multiple thresholds T_{ij}, where j is the number of hops for sending packets as shown in Fig. 3. That means the packets do not need to be transmitted or are only transmitted to its neighbors or within a small number of hops when there is no event occurs and with only Gaussian white noise. The sensors with higher priorities have lower thresholds than sensor nodes with lower priorities. So with the same detected signal strength, the more sensor nodes can receive packets from sensor nodes with higher priority than ones with lower priority, because the sensor nodes with higher priority are more trustworthy.

(3) Using a synchronization scheme, a sensor node senses at different times from its neighbors. The frequency of sensing for every sensor node can be reduced. Some sensor nodes do sampling at the rising edge of a clock and the remainder sensor nodes do sampling at the falling edge of a clock. The phase-locked loop (PLL) technique can be used for synchronization. However, because this is an issue in the analog circuit area, we do not discuss in this paper.

(4) Most of the radio chips used in modern sensors can work at different frequency, so we utilize dynamic sense frequency in our design. When the frequency of the packets it receives increases, the sampling frequency increases up to the maximum frequency. The sampling frequency decreases when the frequency of packets it receives decreases. By this way, the frequency can be reduced most of the time to reduce the energy and bandwidth consumption.

(5) When an event occurs, one or more sensor nodes detect the change first, and then the number of hops for their packets increase. Then the frequency of received packets for nodes near these sensor nodes increases, and these neighbor nodes increases the sensing frequency. Finally, they may also detect the event and increase the number of hops, which forms a "positive feed-back" to let the sensing frequency near the event increase exponentially in order to wake up sensors quickly. The scheme is shown in Fig. 4 and Fig. 5. Fig. 4 shows that when an event occurs, there will be a loop to accelerate the wakeup speed. Similarly, there will be a loop to accelerate falling into the sleep mode, as shown in Fig. 5.

(6) In order to reduce the retransmission and ACK overheads, besides the existing data aggregation algorithms, we propose an event prediction approach. Based on the fact that the physical world is an analog world and thus all measurements such as temperature are continuous and changing relatively slowly. Because the sampling frequency is relatively faster than temperature changes, it is reasonable to predict the next sampling value using the previous sampling values. In other words, the sampling value at t_4 should be close to the extrapolation from the sampling values at t_1, t_2, and t_3. When a sensor node sends a packet to the sink node, it contains not only the current sensing information, but also the delay between this packet and the next packet from the same node, and the prediction of the next packet information. If the prediction is wrong, the second packet with an "Error Flag" will be attached, which will have higher priority than normal packets. When the sink node receives the first packet and does not receive the following packet after a time period based on the prediction time in the packet, the sink node will send a negative ACK packet to the source node. If multiple packets do not arrive at the sink node, especially from the same local area, it is likely that an event happens in that area, and the sink

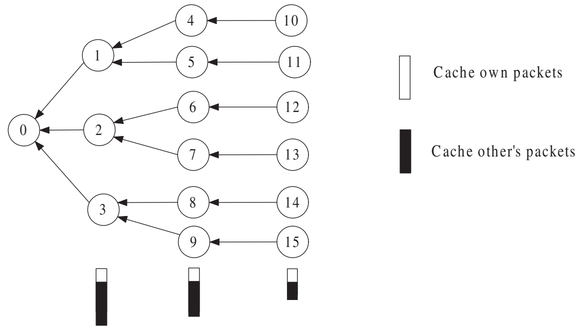

Fig. 6. Cache assignment in active sensor nodes

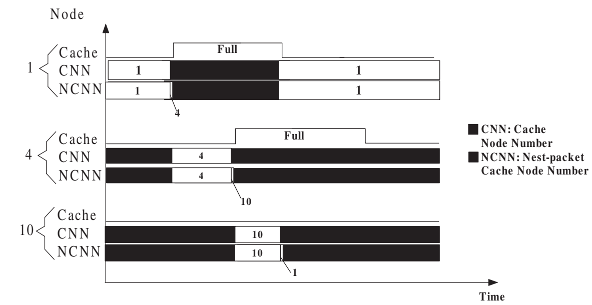

Fig. 7. The cache scheme of the flow 10-4-1-0 in the Fig. 9.

node can make a decision based on the prediction information and packets with the “Error Flag” if it has received any.

The prediction is crucial when the application is a hard real-time system and the link is severely congested; the sink note should make a decision based on the prediction information and the “Error Flag” packets it has received. We use negative ACK packets instead of ACKs because of the fundamental difference between wireless sensor networks and traditional networks. First, the bandwidth in wireless sensor networks is more limited than ordinary networks, and thus reducing the number of ACK packets is more important in WSNs. Using negative ACK packets instead of ACKs can reduce the overall number of packets that need to be sent. Second, because of the prediction of approximate arriving time of the next packet, ACK packets are redundant in the data quality guarantee scheme. Last, the WSNs are more “real-time” than typical networks, so the “send packets to the sink node → does not receive ACK → resend the packet” scheme is not suitable in WSNs.

(7) In order to provide reliable packet delivery, we locate the sensors with more memory near the sink node. The data is cached until expired. The data will be retransmitted if a negative ACK packet is received from the sink node. A cache tag will be attached to the packet, indicating which sensor node should cache the next packet. The sink node keeps the information about all the sensor nodes, including approximate delay from each sensor node to the sink node. The sink node will decide whether it is needed to send the negative ACK packet to let the sensor node retransmit the lost packet. The sink node does not need to send the negative ACK packet if the late packet is not needed any more according to the calculation of round trip time between the sink node and the sensor node.

(8) In order to reduce the out-of-time delivery, the best way is to cache packets near the sink node. We place sensor nodes with more memory near the sink node to store more packets near the sink node. We split the cache memory in each sensor node into two parts: one to store its own packets and another to store received packets from upstream, as shown in Fig. 6.

Each node records the queue information of its downstream nodes. When a sensor node’s cache is nearly full, it will send a packet to its upstream nodes to ask them to store the following packets. When the sensor node has enough cache memory, it will send a packet to the downstream nodes periodically to reopen the cache service. When a sensor node decides to cache a packet, it will mark the cache node number of this packet and the cache node number of the next packet from the same source sensor node. If this sensor node has enough cache memory and the downstream nodes do not have enough cache memory, then the two cache node numbers are identical (this sensor node) and the node reserves the cache memory for the next packet. If a sensor node decides to store this packet and its downstream sensor nodes just have enough cache memory, then the cache node number will be marked as this sensor node, and the next packet cache node number will be marked as the proper downstream sensor node. When this packet arrives at the downstream sensor node, this sensor node will check the next packet cache node number and reserve the cache memory for the next packet if the next packet cache node number matches its own number. If a sensor node decides to store this packet and its downstream sensor nodes and itself do not have enough cache memory to store the next packet, the current packet’s cache node number will be marked as the current sensor node and the next packet cache node number will be marked as the corresponding upstream sensor node number. As we have discussed, the sink node uses the next packet cache node number in the current packet to calculate and decide whether retransmission is needed. Furthermore, the sink node can use the cache node number and the next packet cache node number to estimate the congestion in the links. This scheme is shown in Fig. 7.

A single flow from sensor node 10 to sink node 0 in the Fig. 9 is shown in Fig. 7. We can see that all the packets are cached in sensor node 1 at first. When its cache is nearly full, the sensor node 4 will cache the packets. The CNN and NCNN will be different during the switches. When sensor node 4’s cache is nearly full, sensor node 10 will cache the packets until sensor node 1 has enough free cache memory again.

We also need to consider the situation when a packet is lost before it is cached. For example, when a packet should be cached in cache node 4, the sensor node 10 should cache the packet temporarily after forwarding it to the node 4. Node 4 will send a packet to node 10 to delete this packet after it caches the packet successfully. It is obvious that these added packets with delete information will degrade the overall performance. Fortunately, the real-time requirement from sink node to sensor nodes is not as strict as that from senor nodes to the sink node, so we can combine multiple such packets into a single one, and then the node 10 will delete the cached packets in chunk.

(9) The delay from a selected node to the sink node may vary based on the condition of the link. The sink node can use the following equation to estimate the delay D:

$$D = (1 - \alpha) \cdot D + \alpha \cdot D_{\text{new}} \quad (1)$$

where α is set to 0.125. If a packet does not arrive within this time, it is considered as lost.

(10) We adopt data aggregation ideas such as [17] to save energy. Constructing a tree, especially a minimum one, is expensive and cannot provide good QoS. Hence, we also adopt the multiple-path algorithm such as [14], which is more applicable in real WSNs. Because both approaches have been well-studied, we will not discuss them in this paper and only combine them together to achieve a balance between energy efficient and QoS.

(11) Flow-based routing is more efficient in the event driven sensor networks because most packets have the same destination: sink node. In the Internet, a flow is a string of packets with the same source IP, destination IP, source port, destination port, and type. But the story changes in the wireless sensor networks. We define the flow as follows: a flow is a string of packets generated from sensor nodes in the same local area and following the same path to the sink node after leaving that local area. Given a flow number, a node along the path can easily find out the next hop in its forwarding table. Furthermore, TCAM chips can also be used in the forwarding tables to accelerate the forwarding process.

IV. Analysis

The main goals of the proposed algorithms are to reduce the power consumption of sensor nodes and to improve data quality received by the sink node. We adopted the sleep/active modes idea, but our algorithms differ in that ours focus on how to wake up the corresponding sensor nodes quickly when an event occurs, and reduce power consumption by reducing packet transmissions and adjusting the sampling frequency. We added some information in each packet to reduce the impact of retransmissions and ACKs. In typical networks, senders care more about QoS, e.g., whether the receiver received the packet and retransmission is necessary. We actually moved this task to the receiver side because the sink node is much more powerful than the sensor nodes and the sink node needs to make decisions as soon as possible in many real time applications. In our algorithm, the sink node cares about whether and when the lost packet needs to be retransmitted.

V. Conclusion

In this paper, we propose an energy efficient protocol in event-driven wireless sensor networks to reduce power consumption of sensor nodes and improve data quality received by the sink node. In our algorithm, each sensor node has multiple thresholds and dynamic sampling/hop frequency to save energy when there is no event and wakes up the corresponding sensor nodes quickly when an event occurs. Some information is added to each packet, such as the prediction of the next packet, to reduce the impact of retransmission and ACKs. Tasks are moved from sensor nodes to the sink node, which is more powerful. We plan to test these algorithms in different environments and adopt machine learning methods to modify the parameters such as thresholds in sensors.

References

[1] I. F. Akyildiz, Y. S. W. Su, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] R. Vidhyapriya and P. T. Vanathi, "Energy aware routing for wireless sensor networks," in *Proceedings of International Conference on Signal Processing, Communications and Networking*, Feb. 2007, pp. 545–550.

[3] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM Mobile Computing and Communications Review*, vol. 1, no. 2, pp. 28–36, Apr. 2002.

[4] A. Martirosyan, A. Boukerche, and R. W. N. Pazzi, "Energy-aware and quality of service-based routing in wireless sensor networks and vehicular ad hoc networks," *Annals of Telecommunications*, vol. 63, no. 11–12, pp. 669–681, Dec. 2008.

[5] Y. Dai, Q. Wang, and X. Li, "Mebrs: Energy balancing route scheduling in centralized wireless sensor networks," in *Proceedings of the 1st Asia Symposium on Quality Electronic Design*, 2009, pp. 270–275.

[6] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *Proceedings of 14th IEEE International Workshop on Quality of Service*, Jun. 2006, pp. 83–92.

[7] X. Tang and J. Xu, "Extending network lifetime for precision-constrained data aggregation in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2006.

[8] Q. Han, S. Mehrotra, and N. Venkatasubramanian, "Energy efficient data collection in distributed sensor environments," in *Proceedings of the 24th International Conference on Distributed Computing Systems*, 2004.

[9] A. Silberstein, R. Braynard, and J. Yang, "Constraint chaining: On energy-efficient continuous monitoring in sensor networks," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2006, pp. 157–168.

[10] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2002.

[11] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy efficient and low-latency mac for data gethering in wireless sensor networks," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, 2004, pp. 863–875.

[12] O. Chipara, C. Lu, and G.-C. Roman, "Efficient power management based on application timing semantics for wireless sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, 2005, pp. 361–370.

[13] H. Wu, Q. Luo, and W. Xue, "Distributed cross-layer scheduling for in-network sensor query processing," in *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, 2006, pp. 180–189.

[14] K. Sohrabi and J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[15] G. Zhou, J. Lu, C. Y. Wan, M. Yarvis, and J. Stankovic, "BodyQoS: Adaptive and radio-agnostic QoS for body sensor networks," in *Proceedings of IEEE INFOCOM*, 2008.

[16] R. Iyer and L. Kleinrock, "QoS control for sensor networks," in *Proceedings of IEEE International Conference on Communications*, 2003, pp. 11–15.

[17] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of ACM MobiCom '00*, 2000, pp. 56–67.