Dynamic Routing Algorithm for Priority Guarantee in Low Duty-Cycled Wireless Sensor Networks*

Guodong Sun and Bin Xu

Department of computer science, Tsinghua University, Beijing, China
{sgdcs,xubin}@tsinghua.edu.cn

Abstract. It is a new challenge to provide priority-based delivery in low duty-cycled sensor networks where there are not always-awake communication paths and the wireless links are very time-varying and unreliable. In this paper, we propose a Dynamic Routing Algorithm for priority Guarantee(called DRAG) in low duty-cycled sensor networks. Both schemes of dynamic forwarding decision making and priority-based schedule are used in DRAG to achieve priority guarantee in low duty-cycled sensor networks. We evaluate DRAG via extensive simulations and the results show that DRAG can achieve good performance in delivery ratio and network delay.

1 Introduction

The nodes of wireless sensor networks [5,19] are battery driven and the replacement or recharge of battery is not an easy task for the sensor networks with thousands of nodes embedded physically in a large sensing area. The most important objective of wireless sensor networks is to extend the system lifetime as long as possible.

To bridge the gap between the system lifetime requirement of applications and the constrained energy of nodes, one popular method of increasing longevity of wireless sensor networks is to duty cycle the nodes and let them sleep most of their operation time[17]. The duty cycle of nodes are often set to be 5%, 1%, or less. Even though low-duty-cycling the nodes can improve the system lifetime, the increased longevity comes at the cost of decreased application fidelity: smaller delivery ratio and longer network delay that severely damage the performance of priority-critical applications such as movement tracking and alarm sensor networks. Two features make priority-based routing in low duty-cycled sensor networks more challenging. First, to transmit a packet, the node has to wait a long time until some forwarder wakes up. Second, the wireless link varies with time and is unreliable, leading to frequent retransmissions and impacting the system performance.

* This work is supported in part by the NSF China grant 60803124 and the National Basic Research Program of China(973 Program) grant 2006CB303000.

G. Pandurangan et al. (Eds.): WASA 2010, LNCS 6221, pp. 146–156, 2010.

In this paper, we propose a Dynamic Routing Algorithm for priority Guarantee (called DRAG) in low duty-cycled sensor networks. DRAG aims at more quickly delivering as many high-priority packets as possible. Totally, the main contributions of this study are as follows. First, to the best of our knowledge, this is the first work on the priority guarantee issue in low duty-cycled sensor networks with varying and unreliable links. Second, the proposed algorithm DRAG dynamically selects forwarders in a forwarding candidate set and maintains the link qualities at a low cost, based on which an optimal delay can be achieved. Finally, DRAG improves priority-based delivery in both levels of packet schedule and MAC back-off. Simulation results show that DRAG achieves good priority-based delivery with high delivery ratio and small network delay.

The rest of the paper is organized as follows. Section 2 describes the study background and motivation. Section 3 presents the system model and some related denotations and assumptions. The detailed design is described in section 4 followed by simulations in section 5. Section 6 briefly summarizes some significant work related to this study. We conclude this work in section 7.

2 Background and Motivation

In wireless sensor networks, the nodes are of energy- and bandwidth-constrained. Thus a large amount of work have been proposed to optimize the energy usage in communication, but most of them assume always-available paths, i.e., there are always nodes ready to receive whenever there is data to be transmitted. But to maintain always-awake paths needs frequent idle listening, which consumes significantly the energy of nodes. Raghunathan *et al*[14] found that for most nodes in WINS project, the energy consumption in idle listening mode is almost equal to that in the receiving mode. Additionally, the CC2420 RF chip requires a 19.7mA energy level to perform idle listening, but a 17.4mA energy level to transmit data [2]. Therefore, in order to effectively reduce the energy consumed by idle listening, it is necessary for nodes to turn off their radios most of their time, i.e., operate under a low duty cycle.

However, it is is more challenging to achieve good priority guarantee in low duty-cycled sensor networks. In many priority-critical sensor network scenarios, different data sources may present information with different importance. For example, in a movement-tracking application, the data generated nearby a moving object is more valuable in target localization; in a coal-mine monitoring application, the data describing collapse events obviously ought to be delivered in priori [3]. Even for a general sensor network application, the data incurred by some system functions such as network diagnosis [4] need possibly to be delivered in priority. In those applications, whether the data can be delivered timely and whether the high-priority data can take preference in delivery are very critical to the application fidelity.

We use simulation to investigate the channel contention and priority-based delivery in the low duty-cycled sensor network under two routing algorithms DSF [7] and DSPR(a dynamic shortest-path based routing algorithm which always select the earliest-awake forwarder without considering link qualities). In

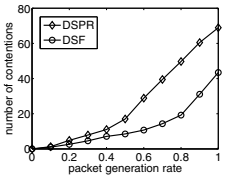


Fig. 1. Number of contention per 100 time units. In simulation, 10 nodes are randomly selected as sources.

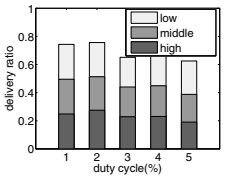


Fig. 2. Delivery ratio of DSPR. In simulation, 3 nodes are randomly selected to generate high-, middle- and low-priority packets, respectively.

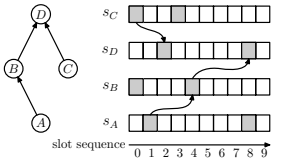


Fig. 3. A low duty-cycled wake-up schedule model. The gray and white blocks are awake and sleep states, respectively.

simulations, 200 nodes are uniform-randomly deployed within a 150×150 square with a single sink centered. The duty cycles of all nodes are 5% and the MAC layer works under a CSMA/CA-like scheme.

The simulation results in Fig.1 show that channel contention can not be neglected even though only 6.6% nodes generate packets in a sensor network of 5% duty cycle because sensor networks are essentially of many-to-one communication. Note that the channel contention under DSPR is more frequent than that under DSF, because DSPR does not consider the link qualities of next-hop, resulting in more retransmissions and then more contention. Fig.2 plots three delivery ratios with high-, middle- and low-priority traffic, respectively, under DSPR(the simulation under DSF obtains similar results). For each duty cycle, those three delivery ratios seem like without obvious differentiation in terms of priorities. Analyzing the simulation results, we find that (1) channel contention is easy to happen to sensor networks even with a very low duty cycle; (2) the varying, unreliable links make it hard to optimize the delivery ratio; and (3) a high-priority packet without a proper control may wait a long time until some forwarder wakes up, impacting the performance of the priority-critical applications. These observations motivate us to design a new routing algorithm for improving the priority guarantee in low duty-cycled sensor networks.

3 System Models

In this study, the duty cycles of nodes are decided directly by the upper layer such as power management or event coverage control [1], rather than by the MAC layer. The priority can be measured in terms of different requirements, such like sensing data type, node location, remaining energy level, and so on. In this study, we generalize the priorities into positive integers. We consider a sensor network consisting of N nodes that are randomly deployed within a square in a uniform distribution.

All duty-cycled nodes can be only in one of two states at a give time: awake or sleeping. In awake states, the node can sense, or receive, or send, or process data. A duty-cycled node can wake up for transmission when it has data ready to be

sent, but it can sense or receive data only in awake states. Specially we assume the sink node is always awake. In this paper, the operation time of nodes are divided into equal-size slots each with a time span of τ(a time unit), and the awake states of each node are periodic due to the periodicity of sensing task requirement. So the node's time can be divided into continuous working schedules. If the length of a single working schedule of a node is 100 slots and contains 10 awake slots, the node's duty cycle is 10%. We assume that for each node, at most one data unit can be transmitted over one hop in a single slot. Fig.3 illustrates a 4-node network where every node is of duty cycle 20% with the working schedule length of 10 slots; the whole awake slots of A is $\{A_1, A_8, A_{11}, A_{18}, \ldots\}$ where A_i means A wakes up at slot i. Due to the periodicity, we abbreviate the awake slots of A as $s_A = \{A_1, A_8\}^*$. We can see that at time 0, node C generates or receives a packet which can be transmitted until time 2 when node D wakes up. The one-hop delay from C to D equals to 2-0=2τ. Similarly, the two-hop delay from A to D is 8-1=7τ. In order to know neighbors' working schedules exactly, local time synchronization is assumed. In this study, we use a CSMA/CA [6] like scheme to control the wireless medium access.

4 Design of DRAG

The basic idea of DRAG is to quickly deliver as many packets as possible meanwhile taking preference to higher-priority packets. DRAG consists of two components: (1) the dynamic selection of forwarders that makes the network deliver packets as soon as possible; and (2) the priority-based packet schedule in both levels of queue management and channel access that offers more chances for higher-priority packets to be transmitted. In DRAG, each node dynamically maintains a set of forwarding candidates, from which the node selects a forwarder in terms of a delay-aware metric, and then schedules a packet with the highest priority. When there are multiple nodes contending a common channel, a priority-based back-off scheme in MAC layer is used.

4.1 Delay-Aware Routing Policy

Determination of Optimal Forwarding candidates. We firstly define by construction the concept of forwarding candidate set of a node and then describe the distributed method of calculating forwarding candidate sets of the node.

Definition 1. Forwarding Candidate Set. First, calculate a union set $\bigcup_v s_v$ where v is any neighbor of u; and then sort ascendantly this union set by the wake-up time of neighbors, and then remove some elements from this union set such that all the remaining wake-up time is not earlier than t. Now the remaining set is called u's forwarding candidate set at time t, denoted as $\mathcal{F}_u(t)$.

For instance, at time t_0 node X has three potential forwarders A, B and C that can relay packets for X; all their duty cycles are 2% where $s_A = \{A_2, A_{40}\}^*$, $s_B = \{B_1, B_{60}\}^*$ and $s_C = \{C_{50}, C_{80}\}^*$. After ascendantly sorting $s_A \cup s_B \cup s_C$

by the wake-up time, we have $\mathcal{F}_X(t_1) = \{B_1, A_2, A_{40}, C_{50}, B_{60}, C_{80}\}^*$. Different from the forwarding sequence defined in [7], our forwarding candidate set allows the repeated selection of a forwarder, e.g., node A is selected two times in $\mathcal{F}_X(t_0)$. Suppose node X wants to transmit a packet to B at time t_1. If this packet is successfully received by B, X completes one transmission; otherwise, X can retry a transmission to some node waking up later than t_1, e.g., to A at time t_2 or C at time t_{50}. Note that a maximum allowed transmission χ is often set by applications in order to save node's energy and bandwidth. Thus we can reduce the infinite set $\mathcal{F}_u(t)$ in Definition 1 into a finite set $\mathcal{F}'_u(t) = \{x_1, x_2, \ldots, x_\chi\}$.

Denote the expected delivery delay of node u as E^d_u. If no considering the unreliable links, for optimizing delay, u selects just a single forwarder v in $\mathcal{F}'_u(t)$ which has the minimum $(d_v + E^d_v)$ among all potential forwarders(d_v is the time for u to wait v's wake-up). But the earliest-awake link is not necessarily the 100% reliable one; the delivery ratios of forwarders impact E^d_u. We use a dynamic programming based method(introduced in [7]) to optimize the expected source-to-sink delay of u; the following equations show the expression of E^d_u

$$E^d_u = \sum_{i=1}^{m} \frac{p_{succ}(v_i)}{E^t_u}(d_{v_i} + E^d_{v_i}) \tag{1}$$

$$E^t_u = \sum_{i=1}^{m} p_{succ}(v_i) E^t_{v_i} \tag{2}$$

where $p_{succ}(v_i)$ is the probability that u (re)transmits to v_i successfully after $i-1$ times failures and E^t_u is the delivery ratio of u. Since the forwarder with a less delay is not necessarily having high link quality, DRAG uses a dynamic programming based method to select an optimal subset $\mathcal{F}^{opt}_u$ from $\mathcal{F}'_u$ such that a minimum E^d_u can be achieved. In order to optimize E^d_u, we have to try every potential forwarder in $\mathcal{F}'_u$ as the last node in the optimal subset, i.e., we have to calculate the expected delay for u with respect to any subset $\{x_1, x_2, \ldots, x_k\}(1 \leq k \leq \chi)$ and then find the minimum expected delay value as E^d_u. For a given subset $\{x_1, x_2, \ldots, x_k\}$, we first initialize a set $\mathcal{F}_u(k)$ with a empty set; we secondly insert x_k into $\mathcal{F}_u(k)$; and next we examine every forwarder from x_{k-1} to x_1. If $x_i(1 \leq i \leq k-1)$ decreases E^d_u, then we insert x_i into the front of $\mathcal{F}_u(k)$; otherwise we examine x_{i-1}. Finally, we select the $\mathcal{F}_u(k)$ resulting in the minimum E^d_u as the optimal forwarding candidate set $\mathcal{F}^{opt}_u$. We will not give the correctness proof due to the space constraint. Note that the expected delay per hop is time-dependent. We calculate a E^d_u for each awake state of a node, because a node can receive data only when it wakes up. Therefore, once a node receives or generates a packet upon an awake slot t, it can calculate $\mathcal{F}^{opt}_u$ according to the E^d values of its forwarders that will wake up later than t.

Since the packet needs not to be relayed again when they arrive at the sink, E^d_{sink} is obviously equal to 0, and $E^t_{sink} = 1$, both of which are the initial conditions of optimizing all nodes' expected delay; and then in DRAG implementation, we use a flooding algorithm based on Bellman-Ford scheme to

distributively calculate E_u^t and then E_u^d for all awake slots of node u. The related implementation details are omitted due to the limited space.

Link-quality Update. Note that the delay optimization in section 4.1 is based on such a fact that every node knows the link qualities of all its forwarding candidates. Consequently we need some link quality updating method to keep the optimal solution. However, updating link quality information needs extra communication overhead. In this section we present a piggyback-based link quality updating method. Technically, when a node receives a packet, it can extract the link quality information in terms of RSSI or LQI or PRR; and then the node can piggyback the link quality information, say LQI in message `ACK`. In this way, almost no extra costs of computation and communication are in need. The link quality update model for node u is shown in Eq.(3)

$$l_{u,v}(t_{i+1}) = (1-\alpha)l_{u,v}(t_i) + \alpha\, l_{u,v}(t_i^{ack}) \tag{3}$$

where $l_{u,v}(t_i)$ represents the link quality between u and v maintained by u, $l_{u,v}(t_i^{ack})$ the new link quality piggybacked in the `ACK` replied by v, $l_{u,v}(t_{i+1})$ the link quality to be used when a next packet will be sent, and α is used to describe the dynamic characteristic of radio links. Parameter α should be function of time. As indicated in [12], the wireless link quality changes continuously over time; larger α means quicker variation of links.

4.2 Priority-Based Schedule

The priority-based schedule which works on both levels of queue management and MAC layer is the other critical technique in DRAG to provide priority guarantee.

Priority-based Queue Management. When a packet is received or generated by a node, it enters the node's buffer queue to wait the chance to be sent. In a FIFO queue, a new-arriving packet is inserted at the end of queue, while the queue's front packet is scheduled to be sent once the channel is seized. The FIFO queue management does not consider and differentiate the priorities of buffered packets. In DRAG, we use a simple, priority-based queue management policy to improve the priority guarantee. When a packet arrives at a node's queue, the node checks the packet's priority to determine an appropriate place relative to the previously sorted packets, i.e., the node always maintains a queue sorted in terms of packets' priorities. Furthermore, a node always selects the highest-priority packet to send as long as the node obtains the channel. If a new-arriving packet makes a node overflow, the node will drop the lowest-priority packet in its queue.

Priority-based Back-off. Besides using the priority-based queue schedule, another way to improve the priority guarantee is adopted in DRAG: the priority-based back-off scheme. Differently than the above queue management, the

priority-based back-off scheme employs the back-off operations in MAC layer to achieve the differentiated transmissions in terms of data priority.

Before starting a transmission, a DRAG node needs firstly to check whether the channel is clear or not. If the channel is busy, i.e., there is data in propagation through this channel, the node can switch immediately to sleep without waiting until the current slot ends, because we have assumed that at most one transmission can be carried out in a slot. If a node, say u determines the channel is clear, it will back off a duration $T_{bk}(u)$ calculated by

$$T_{bk}(u) = \frac{T_{bk}^{max}}{\max\{p_u(i)\}} + t_{rand} \qquad 1 \leq i \leq l_Q(u) \tag{4}$$

where $l_Q(u)$ is the length of u's queue, $p_u(i)$ is the priority value of the i-th packet in u's queue, t_{rand} is a short duration randomly chosen within $[0, T_r]$, and T_{bk}^{max} the maximum back-off time set by applications. By Eq.(4), the node with the highest-priority packet will seize the channel. The parameter t_{rand} is used to avoid multiple simultaneous transmissions of packets with the same priority. Note that the back-off scheme in DRAG delivers high-priority packets in priori but can not avoid entirely the dead lock of low-priority packets in any scenario. One potential solution is to dynamically update the packet priority in delivery according to the travel time or waiting time of packets. But the priority setting upon generation and updating in delivery will be our future work.

5 Simulation

In this section we simulate to evaluate the performance of our work. In our simulation, we focus mainly on two performance metrics as follows: (1) Network Delay. It is measured as the average per-hop delay of a packet traveling for its source to the sink. (2) Priority-differentiated Delivery Ratio. It is measured as the delivery ratio of packets with different priorities. We compare the performance of DRAG with DSF under different duty cycles and packet generation rates.

5.1 Simulation Setup

In simulation, 200 nodes are randomly deployed with a uniform fashion in a $150 \times 150\text{m}^2$ square, and a single sink centers the square. The slot length τ, T_{bk}^{max} and T_r are set to be 20ms, 5ms and 5ms, respectively. and each working schedule is of length 100τ. Each simulation runs $10^5\tau$; and the maximum allowed retransmissions is set to be 4. To simulate faithfully dynamic radio links, we use the CC1000-based radio model described in [20].

It is shown in each simulation that the average number of neighbors per node is about 12 and the minimum-hop count from any node to the sink is 3.5 in average. The simulation under each configuration repeats 20 times with different random topologies and the data is reported with the 95% confidence.

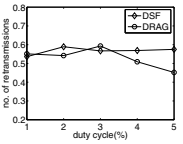


Fig. 4. Retransmissions under different duty cycles

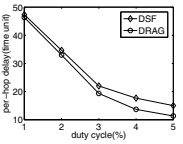
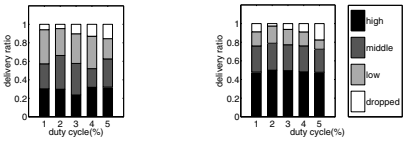


Fig. 5. Per-hop delay under different duty cycles



(a) under DSF (b) under DRAG

Fig. 6. Average delivery ratio under different duty cycles

5.2 Performance Comparison

For DRAG and DSF, we evaluate the priority-differentiated delivery ratio, per-hop delay and retransmissions under different duty cycles and packet generation rates. The energy efficiency of DRAG will be examined in the future work.

Different Duty Cycles. The packet generation rates are all 0.01 in the following simulations. And each source node assigns any packet with one of three priorities(high, middle and low) with the probability of 0.5, 0.3 and 0.2, respectively.

Fig.4 plots the average number of retransmissions of two algorithms under different duty cycles. With the duty cycle increasing, the traffic load increase and then brings more contention. The numbers of retransmissions of duty cycles 4 and 5 under DRAG are both less than those under DSF, because DSF does not immediately update link qualities to maintain a optimal solution. Fig.5 shows that the per-hop delay under each algorithm decreases as the duty cycle of nodes increases from 1% to 5%. And it is seen that DRAG achieves the per-hop delay of 4 time unit smaller than DSF when the duty cycle is 5%. Considering the average delivery hop of 3.5, the network delay under DRAG is 13 time units smaller than that under DSF. These results illustrate the DRAG's advantage of delay over DSF because of the dynamic forwarding candidate selection and link update scheme in use. The simulation in Fig.6 evaluates the performance of priority guarantee under two algorithms. DSF, not using any priority guarantee policy delivers all packets with equal likely, while the stacked histograms in Fig.6(b) show that almost all the high-priority packets generated with probability of 0.5 under DRAG can be sent to the sink. As the duty cycle of nodes increases, the delivery ratio under each algorithm decreases a little. Due to the usage of a low traffic load(0.01 packet generation rate), the total delivery ratio under both algorithms seems similar.

Different Packet Generation Rates. In the simulations with different packet generation rates, let the duty cycles of all nodes be 5%. In each simulation, 20 nodes are randomly chosen as the sources, and each source generates a packet with a probability when it wakes up. The priority assignment is identical to that in section 5.2.

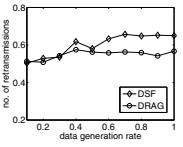


Fig. 7. Retransmissions under different packet generation rates

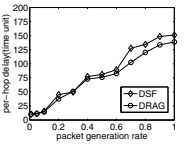
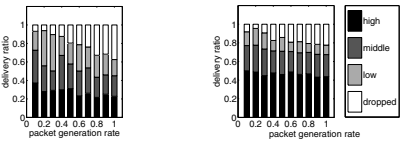


Fig. 8. Per hop delay under different packet generation rates



(a) under DSF (b) under DRAG

Fig. 9. Average delivery ratio under different packet generation rates

Fig.7 shows that the average number of retransmissions under DRAG is slightly less than that under DSF. Fig.8 shows that the per-hop delay under DSF and DRAG both increase as the packet generation rate increases. And when the packet generation rate is beyond 0.6, the per-hop delay under DRAG is clearly smaller than DSF. At the packet generation rate of 0.7, the per-hop delay of DRAG is 32 time units smaller than DSF. Fig.9 compares the priority-based delivery as well as the total delivery ratio. Similar to Fig.6, DRAG also achieves a good priority guarantee in packet delivery. Specially, it can be seen from Fig.9 that DSF drops more packets than DRAG. It is further to be illustrated that the schemes of link update and priority-based packet schedule used in DRAG contribute to the higher total and the priority-based delivery ratio.

6 Related Work

As so far, there are two major kinds of designs for low duty-cycled sensor networks: MAC and routing algorithms. SCP[18] is a MAC scheme designed for ultra-low duty cycle(0.1%) sensor networks. SCP employs LPL(low-power listening) technique and extends the system lifetime of a sensor network by a factor of 3 to 6 times. Kim *et al* [11] investigated the priority-based service differentiation scheme for 802.15.4 sensor networks by adjusting the contention window size and the back-off exponent according to different priority requirement. We will next briefly summarize some major work related to our algorithm.

SPEED [10] includes three types of services: realtime unicast, realtime area-multicast and realtime area-anycast; and it uses end-to-end feedback control and non-deterministic QoS-aware geographic forwarding to achieve realtime communication. Lu *et al*[13] design to reduce the delivery delay by using energy-efficient periodic sleep schedule for sensors. Seada *et al* [15] investigate the geographic routing in sensor networks and show that the product of PRR and the hop distance can be used to select energy efficient paths. Three performance metrics has been proposed in DSF[7]: EDR(expected delivery ratio), EED(expected end-to-end delay) and EEC(expected energy consumption). DSF uses dynamic programming based methods to respectively optimize the above three metrics. We get the heuristic from DSF in performance optimization. But we extend

DSF by employing repeated nodes in forwarding candidate set and introducing a link update scheme, both of which further reduce the per-hop delay. Moreover, DSF examines the cases only with a few sources; the consequent results can not reveal some common communication problems such as channel contention or queue build-up, each of which affects the performances of system and application. Guo *et al*[9] propose an opportunistic flooding scheme for low duty-cycled sensor networks which reduces the transmission redundancy and achieves fast dissemination. Similarly, ADB, another flooding algorithm for asynchronous duty-cycled sensor networks was designed in [16].

7 Conclusions

In this study, we propose a dynamic routing algorithm, called DRAG for improving the priority-guarantee in low duty-cycled sensor networks. This work considers both the low duty cycle of nodes and the varying, unreliable wireless links. In DRAG design, each node forwards packets based on a dynamic forwarding candidate selection and two priority-based schedule schemes. Extensive simulations with a CC1000-based radio model are carried out to evaluate our algorithm. The simulation results show that DRAG achieves good performance over DSF in priority-based delivery ratio and per-hop delay.

References

1. Cardei, M., Thai, M.T., Li, Y., Wu, W.: Energy-efficient target coverage in wireless sensor networks. In: INFOCOM, Miami, USA (2005)
2. CC2420: `http://www.ti.com`
3. Li, M., Liu, Y.: Underground coal mine monitoring with wireless sensor networks. ACM Trans. on Sensor Networks 5, 2 (2009)
4. Liu, K., Li, M., Liu, Y., et al.: Passive diagnosis for wireless sensor networks. In: SenSys, Raleigh, USA, pp. 113–126 (2008)
5. Culler, D., Estrin, D., Srivastava, M.: Overivew of sensor networks. IEEE Computer Magazine (2004)
6. Gast, M.S.: 802.11 wireless networks: the definitive guide, 2nd edn. OReilly, Southeast University Press, Nanjing (2006)
7. Gu, Y., He, T.: Data forwarding in extremely low-duty-cycle sensor networks with unreliable communication links. In: SenSys, Sydney, Australia, pp. 321–334 (2007)
8. Gu, Y., He, T., Lin, M., Xu, J.: Spatiotemporal delay control for low-duty-cycle sensor networks. In: RTSS (2009)
9. Guo, S., Gu, Y., Jiang, B., He, T.: Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In: SenSys, pp. 133–144 (2009)
10. He, T., Stankovic, J., Lu, C., Abdelzaher, T.: Speed: a real-time routing protocol for sensor networks. In: ICDCS, USA, pp. 46–55 (2003)
11. Kim, E.J., Kim, M., Youm, S.K., Choi, S., Kang, C.H.: Priority-based service differentiation scheme for ieee 802.15.4 sensor networks. International Journal of Electronics and Communications 61, 69–81 (2007)
12. Lin, S., Zhang, J., Zhou, G., Gu, L., He, T., Stankovic, J.A.: Atpc: adaptive transmission power control for wireless sensor networks. In: SenSys, pp. 223–235 (2006)

13. Lu, G., Sadagopan, N., Krishnamachari, B., Goel, A.: Delay efficient sleep scheduling in wireless sensor networks. In: INFOCOM, Miami, USA (2005)
14. Raghunathan, V., Schurgers, C., Park, S., Srivastava, M.: Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine 19, 40–51 (2002)
15. Seada, K., Zuniga, M., Helmy, A., Krishnamachari, B.: Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In: SenSys, Baltimore, USA, pp. 108–119 (2004)
16. Sun, Y., Gurewitz, O., Du, S., Tang, L., Johnson, D.B.: Adb: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. In: SenSys, pp. 43–56 (2009)
17. Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordination adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. on Networking 12, 493–506 (2004)
18. Ye, W., Silva, F., Heidemann, J.: Ultra-low dudty cycle mac with scheduled channel polling. In: SenSys (2006)
19. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Computer Networks 52, 2292–2330 (2008)
20. Zuniga, M., Krishnamachari, B.: Analyzing the transmissional region in low power wireless links. In: IEEE SECON, pp. 517–526 (2004)