ORAS: Opportunistic Routing with Asynchronous Sleep in Wireless Sensor Networks

Shucheng Liu
Department of Computer Science
University of Science and Technology of China
Hefei, China
Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
willyliu@mail.ustc.edu.com

Mo Sha
Department of Computer Science and Engineering
Washington University in St. Louis
USA
ms31@cec.wustl.edu

Liusheng Huang
Department of Computer Science
University of Science and Technology of China
Hefei, China
lshuang@ustc.edu.cn

***Abstract*—Opportunistic routing and asynchronous sleep are both recent techniques those change the traditional way of routing and MAC layer implementation. The existing opportunistic routing schemes, designed for increasing throughput of connection rather than saving energy, are not appropriate for Wireless Sensor Networks due to low energy efficiency caused by overhearing and scheduling. In this paper, we propose *opportunistic routing with asynchronous sleep* (ORAS), a robust integrated routing and MAC protocol that combines opportunistic routing and asynchronous sleep. ORAS makes a sender rendezvous with just enough asynchronous sleeping neighbors to ensure the packet received by a potential forwarder with a high probability. ORAS is opportunistic in nature as it exploits the spatial and temporal diversity of data traffic to do transmission. We evaluate the efficiency of ORAS through extensive experiments on testbed of TelosB motes. Our results show that ORAS can achieve high performance in terms of packet delivery ratio, latency and power efficiency.**

Keywords-wireless sensor network; opportunistic routing; asynchronous sleep

I. INTRODUCTION

Wireless sensor networks (WSNs) are becoming widely used in many areas, e.g., environmental monitoring and motion monitoring [1]. In a large scale WSN with many battery-powered sensor motes deployed in a wide area, data is usually delivered by multi-hop along a route from the source node to destiny [2][3]. Therefore energy efficiency and communication delay are two fundamental metrics in evaluating the sensor network communication protocols.

Multi-hop routing algorithms of wireless networks typically resemble the ones used in wired networks. In such approaches, the best route based on certain cost metric, e.g., link quality, is first chosen from the source to the destination. Such unicast-based approaches, however, do not take advantage of broadcast nature of wireless communication and can result in high routing cost and energy consumption. As wireless links are inherently lossy, transmission failures or retransmissions are pervasive, resulting in significant energy wastage and communication quality degradation.

Recently, opportunistic routing [4] has emerged as a novel routing scheme that can effectively improve the capacity of multi-hop wireless networks by taking advantage of broadcast nature of wireless communication. In contrast to traditional unicast routing where the next-hop forwarder is designated before the transmission, opportunistic routing defers the decision of forwarder until after the transmission occurs. Such deferred decisions give a packet more opportunities to make progress toward the destination, e.g., by traveling through long but lossier communication links that the traditional unicast routing would avoid [4]. Furthermore, opportunistic routing is more robust to node or link failures because the success of a packet delivery does not depend on any particular node or link. However, the existing opportunistic routing schemes [4], which are designed for wireless workstations with little resource constraints, are ill-fit with energy constrained multi-hop wireless sensor networks.

To achieve high energy efficiency and low communication delay for WSN, we propose a novel communication protocol, called Opportunistic Routing with Asynchronous Sleep (ORAS) for WSNs. ORAS integrates two components, a lightweight opportunistic routing

978-1-4244-5824-0/$26.00 ©2010 IEEE

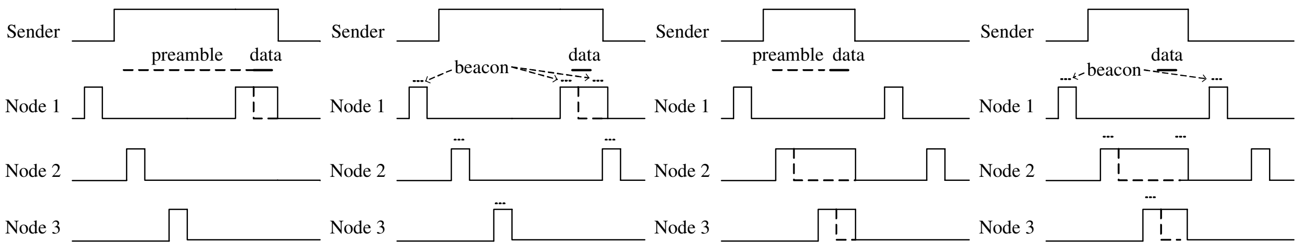

Figure 1. Wake-up processes of traditional unicast routing and ORAS: (a) Wake-up process of traditional unicast routing with X-MAC; (b) Wake-up process of traditional unicast routing with RI-MAC; (c) Wake-up process of X-MAC based ORAS; (d) Wake-up process of RI-MAC based ORAS. Node 1 is the next-hop receiver in the traditional unicast routing in (a) and (b).

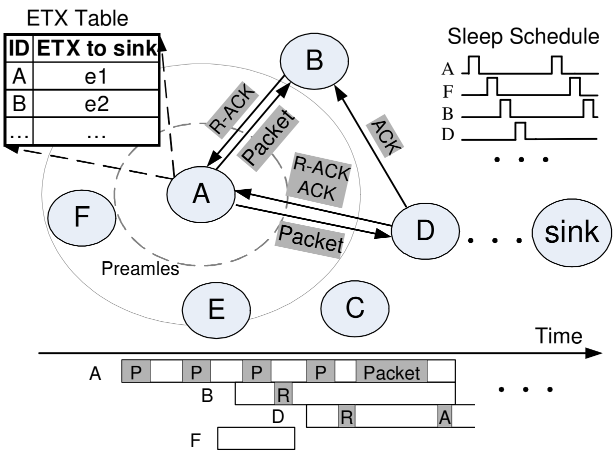

Figure 2. Overview of ORAS design.

component and a novel wake-up windows schedule component. ORAS employs an efficient per-packet forwarder arbitration scheme in which nodes that received a packet competes to be the next-hop forwarder of the packet based on their expected number of transmissions to the base station. To reduce energy consumption, each node in ORAS independently maintains its own low duty cycle with little inter-node coordination or time synchronization. Such an asynchronous sleep schedule scheme avoids the overhead of maintaining consistent duty cycle or synchronizing local times among neighboring nodes. Before each packet transmission, ORAS wakes up just enough sleeping neighbors to ensure the packet will be received by a potential forwarder with a high probability. The novelty of ORAS lies in its effective integration of asynchronous sleep schedule with opportunistic routing, which achieves a good tradeoff between communication delay and energy consumption. First, asynchronous sleep schedule mitigates the impact of nodes' sleeping on the communication delay of opportunistic routing because it allows a sender to wake up multiple potential forwarders promptly irrespective low node duty cycles, which would not be possible in synchronous sleep schedule. Second, opportunistic routing reduces the energy consumed by neighbor discovery, because it only requires a subset of neighbors of the sender to be woken up. Our major contributions are summarized as follows.

- We develop ORAS, a robust integrated routing and MAC protocol that combines opportunistic routing and asynchronous sleep for WSNs.
- We evaluate the efficiency of ORAS in a 10-node testbed of TelosB motes. Our implementation is in TinyOS-2.0.2. Our results reveal that ORAS can significantly increase energy efficiency and reduce communication delay compared with traditional unicast routing.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the design and implementation of ORAS. Section 4 presents the experimental results and Section 5 concludes the paper.

II. Related Work

Energy-constrained wireless sensor nodes suffer from idle listening. Many approaches [5][6] using duty cycling are proposed recently to solve the idle listening problem. In these works, sensor nodes reduce energy consumption by alternatively turn their radio on and off periodically. For example, with a 10% duty cycle, a node has its radio on only 10% of the time and sleep for rest of time, resulting in substantial energy savings. However, radio communication efficiency between energy-constrained sensor devices is affected by duty cycling due to asynchronous sleep scheduling. A disadvantage of asynchronous sleep schedule is its high neighbor discovery and wake-up costs. As each node maintains its own duty cycle schedule, a transceiver needs to discover its neighbors' sleep schedules and wake up them if necessary before each packet transmission.

In the existing asynchronous approaches, such as B-MAC [5], X-MAC [7] and RI-MAC [6], are less efficient in latency and power efficiency due to long pending time of transceiver. In B-MAC and X-MAC, which employ low power listening (LPL), a sender transmits a preamble lasting at least as long as the sleep period of the receiver, resulting in wasteful energy consumption. Furthermore, the preamble transmission occupies the wireless medium for a long time until DATA is delivered, leading to inefficient latency in the condition of contending traffic flows. RI-MAC solves these problems by proposing a receiver initiated transmission to duty cycle MAC protocols, in which senders wake up and wait for request beacons coming. However, these protocols lead to supernumerary latency and power consumption as they are mainly optimized for light traffic load in which the duty cycle is low, e.g., 15% or less. The wake-up processes of traditional unicast routing during a packet transmission with X-MAC and RI-MAC are illustrated in Fig. 1(a) 1(b). The dotted segments represent the original sleep schedule. In 1(a), the preamble must last at least a sleep period to ensure Node 1 will be woken up. Node 2 and 3 are woken up although they are not the receiver. There is no long preamble in 1(b); however, the long pending time of sender is almost same long as in 1(a). All these problems lead to energy wastage and high latency. Such an overhead can significantly shorten nodes' lifetime [8]. On the contrary, the preamble length of ORAS in 1(c) and the pending time of sender in 1(c) 1(d) are much shorter because sender only needs to wake up enough (two in this example) potential forwarders.

III. Design of ORAS

In this section, we propose our Opportunistic Routing with Asynchronous Sleep (ORAS) design. The objectives of ORAS are: 1) to achieve better energy efficiency by utilizing segmented preambles and the feature of opportunistic routing, and 2) to increase the throughput of sensor networks while causing no more latency by utilizing ETX based forwarder election and Request-ACK (R-ACK).

Fig. 2 illustrates the main idea of ORAS based on an X-MAC protocol. Each node maintains a table of ETX to the base station (E_b), which is obtained in neighbor probing phase, and periodically wakes up based on its own sleep schedule, as shown above the topology. The neighbor table contains E_b values of both the node itself and neighbor nodes for the ETX based forwarder election. In X-MAC based ORAS, a node with pending data to send, e.g., node **A** in the figure, stays active and initiates to send segmented preambles containing its E_b value with short intervals while waiting for the Request-ACK (R-ACK) from the potential receivers, as shown in the time sequence of wakeup time and packet transmissions below the topology. The wakeup time is marked with white strip below the time label and the grey blocks represent the transmissions in air, i.e., segmented preamble, R-ACK and ACK. When a neighbor node of **A** wakes up and hears the preamble, it looks at the E_b included in the packet. If that node has a larger E_b, e.g., node **F** in the figure, it switches back to sleep mode immediately and continues its duty cycling as if the medium had been idle. As seen in the figure, node **F** quickly returns to sleep, thus avoiding the overhearing problem. If that above-mentioned node has a smaller E_b, e.g., node **B** and **D**, it is regarded as a candidate for recipient. The candidate sends an R-ACK containing its E_b value to the sender in next interval of segmented preambles and remains awake for the subsequent data packet. After node **A** collects enough candidates, it starts its DATA transmission immediately, which will be acknowledged by an ACK beacon from **D** as discussed later. Upon receiving the ACK, **B** returns to sleep and so does *A* if there is no more data to transmit. Eventually, **D** owns the packet and will start the transmission phase as **A** did.

As illustrated in the previous example, ORAS significantly reduces overhearing, the amount of time a pair of nodes occupy the medium before they reach a rendezvous time for data exchange and sender pending time, compared to the unicast routing based on B-MAC, X-MAC and RI-MAC. ORAS enables more contending nodes to exchange DATA frames with their intended receivers, which helps to increase capacity of the network and thus the potential throughput.

A. Lightweight Opportunistic Routing

We now describe how ORAS conducts lightweight opportunistic routing. The basic idea of lightweight opportunistic routing is to perform the forwarder arbitration process on the per packet basis. A key advantage is that a node does not need to keep track of inter-packet reception status. In addition, it reduces the end-to-end communication delay when the network workload is low compared with the bath mode used by previous work [4]. The opportunistic routing component of ORAS compromises the following two phases.

1) Periodic Neighbor Probing Phase

To do opportunistic routing, each node needs to know the cost to transmit a packet to the base station. Here we use the expected number of transmissions (ETX) [9] to the base station E_b of its neighbors to define the cost metric, which is equal to the sum of ETXs along the shortest path to the base station. A similar metric is used in a prior work [4].

To obtain E_b, each node in ORAS records the PRR of the link to every neighbor and maintains a neighbor table containing (ETX) to the base station as shown in Fig. 2. In the probing phase, the E_b value is obtained by periodically broadcasting the ETX value and the neighbor table is generated or updated after receiving the value of others. ORAS conducts the probing phase infrequently as the performance of opportunistic routing is not sensitive to any particular set of links, which is contrast to the unicast routing [10][11] in which fresh link quality information is needed to ensure good performance. This feature of opportunistic routing significantly reduces the overhead of link profiling which is particularly desirable for energy-limited WSNs.

2) Forwarder Election Phase

In the forwarder election phase, ORAS makes a sender rendezvous with just enough asynchronous sleeping neighbors to ensure the packet will be received by a potential forwarder with a high probability. ORAS achieves this by utilizing different tactics based on varied state-of-art MAC protocols, such as X-MAC [7] and RI-MAC [6].

The forwarder election of an X-MAC based ORAS is as follows. A node with pending data to send stays active and starts to send segmented preambles containing its E_b value with short intervals while waiting for the Request-ACK (R-ACK) from the potential receivers. When a neighbor node of **A** wakes up and hears a segmented preamble, it looks at the E_b that is included in the preamble. If the node has a larger E_b, it returns to sleep immediately and continues its duty cycling as if the medium had been idle; if the node has a smaller E_b, it is regarded as a candidates for recipient. The candidates send an R-ACK containing its E_b value to the sender in next interval of segmented preambles and remain awake for the subsequent data packet. To count the number of candidates for recipient, ORAS implements a Candidates Counter (CC) in senders to obtain candidates number by counting the R-ACKs received. If CC reaches the election

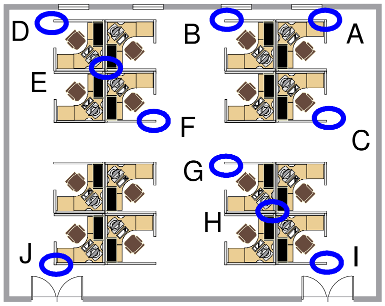

Figure 3. A small indoor testbed in office: The motes are marked with blue ellipse, **A** is the source mote and **J** is the sink mote.

number, which is specified by the user, the sender includes the candidates ID list sorted by E_b in its packet and transmits it immediately. Upon receiving a packet from the sender, candidates look at the Candidate List (CL) and start the forwarding timer according to the following equation,

$$t_f = i \cdot T_{ACK}, \quad (1)$$

where i is the order of candidate in CL and T_{ACK} is the time for ACK transmission. When the forwarding timer fires, the candidate broadcasts an ACK message and starts another transmission phase as sender. Note that this ACK beacon's role is threefold:

- Acknowledging the sender successful reception of the transmitted packet and suppressing retransmissions.
- Announcing the right to forward the packet to the next hop among all candidates for recipient.
- Inviting a new DATA frame transmission to the same receiver.

Upon receiving the ACK, other candidates returns to sleep, and so does the sender if there is no more pending data to transmit. The ORAS based on RI-MAC is similar to the one based on X-MAC, except that there is no preamble in RI-MAC and the E_b value is only included in R-ACK.

The retransmission is an option of ORAS. In retransmission mode, a sender implements a rebroadcasting timer and process according to it. The sender starts a rebroadcasting timer after broadcasting the packet. If there is no ACK received before the rebroadcasting timer fires, the sender will rebroadcast the packet.

B. *Wake-Up Windows Schedule*

As shown by Fig. 1(c) 1(d), the key to reducing the delay of a packet transmission is waking up enough neighbors promptly. This is challenging since neighbors maintain their sleep schedules independently and it would be costly for the sender to discover each individual neighbor's schedule. Our solution is to distribute the wake-up windows of nodes in the neighborhood uniformly such that each node will incur the same delay of waking up its neighbors. We employ a slotted scheme in which the sleep period is divided into equal slots and each node randomly chooses a slot to turn on its radio. Suppose the sleep period lasts for S seconds and the duty cycle of each node is $0 < c \leq 1$. The duty cycle determines the lifetime of the network and is often specified by the user. The sleep period is divided into $l = ceil(1/c)$ slots. Note that the duration of sleep period must be chosen to ensure that a slot is long enough to receive a packet. Initially, each node randomly chooses a slot from the l slots to turn on its radio, as shown in Fig. 1. The wake-up slots of different nodes may get synchronized over time due to clock drifts, resulting in non-uniform distribution. To account for this issue, each node may periodically change its slot.

IV. EXPERIMENTAL EVALUATIONS

To study the performance of ORAS, we experimentally analyze the energy consumption, latency, and throughput of ORAS using high fidelity sensornet testbeds. In what follows, we first present our experimentation methodology, and then discuss our measurement results.

A. *Methodology*

We evaluated ORAS in an implementation of a small portable testbed network of TelosB [12] motes running TinyOS [13]. We have used the portable testbed for both indoor and outdoor experiments, and have observed similar phenomena. Due to the limitation of space, here we only present the data for the case when the portable testbed is deployed in an office as shown in Fig. 3. The packet size is set to 28 bytes, as the default packet size value in the UPMA package [14]. All experiments are repeated for ten separate rounds.

To evaluate performance of ORAS, we compared ORAS against unicast routing with state-of-art MAC protocols, i.e., X-MAC and RI-MAC. We create a data transmission scenario as follows. As shown in Fig. 3, the source mode **A**, **B** and **C** generate *m* packets per 1 seconds, where *m* is a tunable number, with the *m* packets uniformly distributed over each 1 seconds time window, and these packets need to be delivered over multi-hop paths via potential forwarder motes **D**, ... , **I** to the sink mote **J**. At the beginning of each experiment run, each mote randomly chooses its wakeup time between 0 and 1 second. Thus, the wakeup/sleep schedule of each mote is randomized and varied at each experiment run. We do this because the experimental results show that the time schedule of the forwarder can significantly influence the performance of unicast routing. When implementing the unicast routing, the source choose the forwarder base on ETX value, e.g., in one of our experiment runs, mote **E** is chosen as the forwarder because it has the smallest ETX among all the potential forwarder motes.

We study the performance under different scenarios by varying the traffic load and the duty cycle. The traffic load is varied by varying the number of source motes and the duty cycle is varied from 5% to 20% by controlling the length of each mote's period awake. Due to the limitation of space, here we only present the data for the case when the duty cycle is 10%.

To evaluate the energy consumption performance, we implement a cumulative awake timer in each mote. The cumulative awake time of each mote is transmitted to the sink mote at the end of each experiment run. The energy

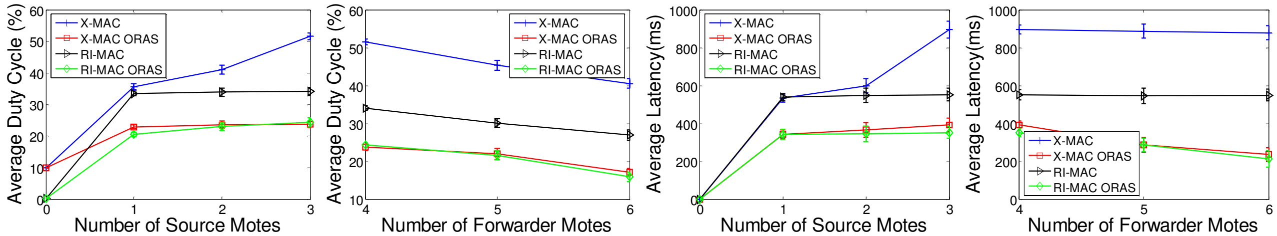

Figure 4. Average Duty Cycle and Average Latency: (a) 4 forwarder motes; (b) 3 source motes; (c) 4 forwarder motes; (d) 3 source motes.

consumption is evaluated by effective duty cycle. We do this because energy consumption of different radios varies significantly [15], and this method has been widely used by previous works [6][7]. To obtain the average latency of received packets, we synchronize the source mote and the sink mote at the beginning of each experiment run and implement a timer on both of them. We include in each packet generated by the source mote a time stamp, which indicates its birth time, so that the sink mote can calculate the latency of received packets.

B. *ORAS Performance*

Fig. 4 shows the average duty cycle of motes and latency when experiments are conducted using varied protocols, varied number of source and forwarder motes, respectively. Error bars show the 95% confidence interval. Notice that the y labels in Fig. 4(a) 4(b) represent the average duty cycle of all the nodes. From the experimental results we observe that the duty cycle of some nodes are even higher than 70% and 50% in unicast routing with X-MAC and RI-MAC, separately, when there are 3 source motes transmitting. Fig. 4(c) 4(d) show that the latency of unicast routing with X-MAC and RI-MAC will be higher than 800ms and 500ms when there are 3 source motes transmitting. These figures show that ORAS significantly reduces the average duty cycle of motes and latency when there are multiple sources or forwarders. ORAS achieves this by greatly reducing 1) the congestion backoff time of X-MAC caused by waiting for the medium to be clear and 2) the pending time of RI-MAC for rendezvous a sender and a receiver.

V. CONCLUSION

In this paper, we propose opportunistic routing with asynchronous sleep (ORAS), a robust integrated routing and MAC protocol, which combines opportunistic routing and asynchronous sleep. ORAS achieves better energy efficiency than traditional unicast routing, especially when duty cycle is low, and increases the throughput of sensor networks while causing no more latency. We implemented ORAS in TinyOS-2.0.2 and conducted extensive experiments on a 10-node testbeds of TelosB motes. Our experimental results show that ORAS can significantly increase energy efficiency and reduce communication delay compared with traditional unicast routing.

REFERENCES

[1] D. CRULLER, D. Estrin, and M. Srivastava, "Overview of sensor networks," Computer(Long Beach, CA), vol. 37, pp. 41-49, 2004.

[2] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," presented at the Proceedings of the 9th annual international conference on Mobile computing and networking, San Diego, CA, USA, 2003.

[3] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," SIGCOMM Comput. Commun. Rev., vol. 34, pp. 133-144, 2004.

[4] S. Biswas and R. Morris, "ExOR: opportunistic multi-hop routing for wireless networks," SIGCOMM Comput. Commun. Rev., vol. 35, pp. 133-144, 2005.

[5] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004.

[6] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," presented at the Proceedings of the 6th ACM conference on Embedded network sensor systems, Raleigh, NC, USA, 2008.

[7] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," presented at the Proceedings of the 4th international conference on Embedded networked sensor systems, Boulder, Colorado, USA, 2006.

[8] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004.

[9] D. S. J. D. Couto, "High-throughput routing for multi-hop wireless networks," Massachusetts Institute of Technology, 2004.

[10] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," presented at the Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2, 2005.

[11] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," presented at the Proceedings of the 12th annual international conference on Mobile computing and networking, Los Angeles, CA, USA, 2006.

[12] Crossbow Telos B motes. http://www.xbow.com.

[13] Tiny OS. http://www.tinyos.net.

[14] K. Klues, G. Xing, and C. Lu, "Link layer support for unified radio power management in wireless sensor networks," presented at the Proceedings of the 6th international conference on Information processing in sensor networks, Cambridge, Massachusetts, USA, 2007.

[15] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," presented at the Proceedings of the 4th international conference on Embedded networked sensor systems, Boulder, Colorado, USA, 2006.