Probabilistic Polling for Multi-Hop Energy Harvesting Wireless Sensor Networks

Zhi Ang Eu and Hwee-Pink Tan
Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore
Email: {zaeu,hptan}@i2r.a-star.edu.sg

Abstract—**In this paper, we propose a medium access control protocol (EH-MAC) for multi-hop energy harvesting wireless sensor networks (EH-WSNs). In EH-WSNs, the main goal is to match energy consumption with the energy harvesting rate which is different from minimizing energy consumption in battery-operated WSNs. Unlike most existing MAC protocols that are designed to extend network lifetime, EH-MAC is designed to achieve high throughput given the varying amount of ambient energy that can be harvested from the environment at different locations and times. EH-MAC is based on asynchronous, receiver-initiated polling; however, unlike traditional random backoff mechanisms, it uses probabilistic polling to reduce data packet collisions. EH-MAC also dynamically adjusts the number of polling packets to minimize interference. Performance results show that EH-MAC increases network capacity and data throughput over other MAC protocols for EH-WSNs.**

I. INTRODUCTION

Recent advances in energy harvesting technologies [1] have made it possible for sensor nodes to rely solely on energy harvesting devices for power. Each energy harvesting wireless sensor node typically comprises one or more energy harvesters, an energy storage device (e.g., supercapacitor) to store the harvested energy, a sensor for measurement, a microcontroller for processing and a transceiver for communications. However, there remains many open research problems in EH-WSNs [2] including the highly variable and often unpredictable energy harvested from the environment as well as the limited transmission range of each node. Furthermore, since each node can operate as long as ambient energy is available, balancing energy usage with the amount of energy harvested is the key objective in protocol design for EH-WSNs.

In this paper, we propose EH-MAC (Energy Harvesting MAC) that can achieve high throughput and fairness in a multi-hop EH-WSN using a probabilistic polling mechanism that adapts to changing energy harvesting rates or node densities to manage packet collisions and channel contention. The rest of the paper is organized as follows: Section II describes related work on MAC protocols for WSNs. The key features of our proposed EH-MAC scheme are described in Section III. Performance evaluation results are presented in Section IV, while conclusions are given in Section V.

II. RELATED WORK

MAC protocols for WSNs can be classified under *scheduled*, *random access* and *polling* schemes. In scheduled MAC protocols (e.g, [3]), time slots are assigned for each node to transmit so that idle listening can be eliminated, and collision can be avoided. However, the exchange of time schedules incurs additional overhead and requires a time synchronization protocol. Since the energy source is unpredictable in EH-WSNs, it is difficult for nodes to exchange time schedules since they do not know future energy availability. However, a Wakeup Schedule Function (WSF) [4] can solve this problem by allowing each node to wake up asynchronously, i.e., without coordination with other nodes. The WSF is defined using a (u, w, v) block design, where each node is awake over a block of u slots, and is active over w slots such that any two nodes would have at least v overlapping active slots.

Random access protocols do not need to exchange schedules but incur additional (i) idle time for the node to sense the channel before transmitting, and (ii) overhearing time to listen to packets not destined to itself. In sender-initiated protocols, packet transmissions are initiated by the sender node. In B-MAC [5], the sender transmits preamble symbols, and the receiver uses an adaptive preamble sampling scheme with low power listening to reduce duty cycle and minimize idle listening. X-MAC [6] improves upon B-MAC by using shorter preambles and reducing energy consumption for receivers. In receiver-initiated protocols (e.g., RI-MAC [7]), data transmissions are initiated by a beacon packet from the receiver.

Polling is a form of receiver-initiated protocol that relies on the request-data-acknowledge mechanism used in many data dissemination protocols (e.g., SPIN [8]). Here, the receiver polls the sender based on its node identity for data transmission. However, in EH-WSNs, the receiver may not know which node(s) is/are awake at the instant of polling due to the unpredictability in energy harvesting process. Hence, probabilistic polling was proposed for *single-hop* EH-WSNs in our previous work [9]. Instead of identity-based polling, the data sink sends a *contention probability*, for which active nodes respond to accordingly. This contention probability is dynamically adjusted at the sink according to the network load (determined by the network density and energy harvesting rates). It is shown to be effective in resolving contentions by achieving high throughput while maintaining good fairness for single-hop EH-WSNs. Hence, in this paper, we extend probabilistic polling to *multi-hop* scenarios, which is non-trivial as (i) the transmission of polling packets from multiple nodes need to be coordinated, (ii) the contention probability at each node must be adjusted in a distributed manner and (iii) the hidden terminal problem exists and needs to be resolved.

III. ENERGY HARVESTING MAC (EH-MAC)

A. Energy Model

Even with the state-of-the-art energy harvesters, the rate of energy harvesting is much lower than typical power consumption levels in a wireless sensor node. As such, each EH-WSN node can be in one of two states: (i) *charging* - in this state, the node is inactive and harvested energy is cumulatively stored; (ii) *active* - in this state, there is sufficient stored energy to operate the node while it continues to harvest energy from the environment. We consider a simple energy management scheme whereby the node switches to active state whenever it has sufficient energy, E_m, to remain active for t_a, after which it switches back to charging state, as illustrated in Fig. 1.

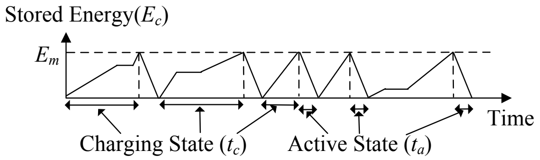


Fig. 1. Simple Energy Management Scheme

The required stored energy, E_m, can be expressed as $E_m = \max(P_{rx}, P_{tx})t_a$, where P_{rx} and P_{tx} are the receive and transmit powers of the node respectively, and t_a is the active time in each wakeup cycle. It can be expressed as $t_a = n_a t_{tx}, n_a > 1$, where $t_{tx} = 8s_d/\alpha$ is the transmission time for a data packet of size s_d bytes with a transmission rate of α bps and n_a is a system parameter (set to 20 in this paper).

B. Probabilistic Polling

At the start of the active period, after some random time, a node will send a polling packet (of size s_p bytes) to request data packets (of size s_d bytes) from other nodes if the channel is clear. Otherwise, the node will wait until the channel is clear before repeating the process. This random time, chosen between 0 to t_{max} (set to t_{tx} in this paper), is to enable nodes to listen to other polling packets as well as to randomize access to the wireless channel so as to reduce the probability of collisions between polling packets.

To reduce the number of data packet collisions, each receiver node maintains a contention probability, p_c, which is used in the polling packet to indicate the probability that a sender node should transmit its data packet. Upon receiving a polling packet, a node would generate a random number $x \in [0, 1]$: it transmits its data packet if $x < p_c$; otherwise, it will defer transmission for t_{tx} to avoid possible collision with the data packet sent by another active neighbor in response to the polling packet. A node will not transmit a polling packet if its buffer, of size s_b packets, is full.

C. Contention Resolution in Probabilistic Polling

In [9], we have shown that the optimal p_c that maximizes throughput is $1/n_{active}$, where n_{active} is the number of active neighbors of the node that sent the polling packet. Therefore, we propose two different dynamic contention probability adjustment schemes to achieve this value:

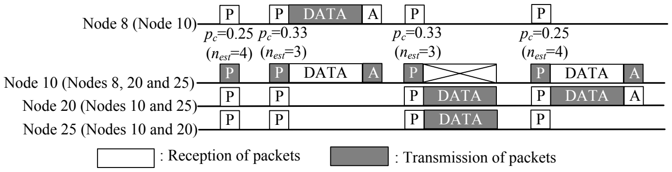
*1) **AIMD**:* The first method is to adjust the contention probability *directly* using an Additive-Increase Multiplicative-Decrease (AIMD) algorithm, where p_{lin} and p_{md} are the additive increase and multiplicative decrease factors respectively. We use $p_{lin} = 0.01$ and $p_{md} = 0.5$ as we have shown in [9] that these values give high throughput for single-hop scenarios.

*2) **ENAN**:* The second method, which is not explored in [9], is to adjust the contention probability using $p_c = 1/n_{est}, n_{est} \geq 1$, where n_{est} is the *E*stimated *N*umber of *A*ctive *N*eighbors (ENAN). A neighboring node is considered an active neighbor if it is in the active state and can respond to polling packets. ENAN can be more accurate than AIMD when the average number of active neighbors is not high. A node only estimates the number of active neighbors but does not need to know their identities thereby eliminating the use of costly neighborhood discovery schemes. The value of n_{est} for the i^{th} polling packet depends on the outcome of the $(i-1)^{\text{th}}$ polling packet: if exactly one node responded, n_{est} may have been estimated correctly; if multiple active neighbors responded, n_{est} is increased by 1 as it may have been underestimated; if there was no response, n_{est} is decreased by 1 (subject to a minimum value of 1) as it may have been overestimated. A packet transmission outcome classifier [10] can be used to differentiate between packet losses due to collisions or weak signals.

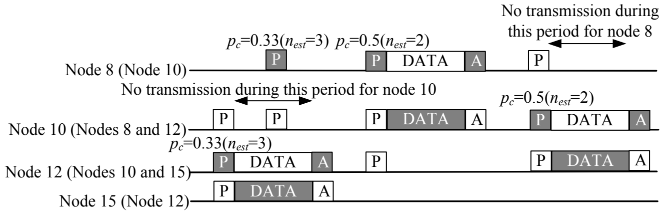
Fig. 2a illustrates the process of receiving data packets from other nodes where the neighbors of each node are indicated in the brackets, and the shaded and unshaded boxes represent packet transmission and reception respectively. When node 10 is in the active state, it will turn on its transceiver to listen to the channel. Assuming that n_{est} is 4 for node 10, when the channel is clear, it will send a polling packet with $p_c = 0.25$. As no response is received, it reduces n_{est} to 3 and retransmits another polling packet with $p_c = 0.33$ after a random interval. After node 10 has received a data packet from node 8, it will send an acknowledgement (ACK) packet (of size s_{ack} bytes) to node 8. After some random time, the node will send another polling packet with the same p_c. If it receives concurrent transmissions from multiple nodes resulting in a corrupted packet, it will increase n_{est}. If a received data packet needs to be forwarded, it will be stored in the buffer.

Fig. 2b illustrates the process of sending data packets. When node 10 receives a polling packet from node 12 but decides not to transmit, it will defer transmission for a period of one data transmission to avoid potential collision at node 12. When node 10 receives the next polling packet from node 8, it decides to transmit a data packet. When the data packet is successfully sent to node 8, node 10 will receive an ACK and remove the data packet from its buffer. If it does not receive the ACK due to wireless losses, the data packet will not be removed from the buffer. A node can receive and transmit data packets in the same cycle.

The AIMD scheme is more conservative than ENAN scheme: p_c decreases much faster in AIMD than ENAN when



(a) EH-MAC receiver example for node 10



(b) EH-MAC sender example for node 10

Fig. 2. Description of EH-MAC(ENAN)

collisions are detected, but increases slower in AIMD than ENAN when a node receives no response from its neighbors. Algorithm 1 summarizes both methods. We use EH-MAC to refer to both EH-MAC(ENAN) and EH-MAC(AIMD).

Algorithm 1 Updating contention probability p_c in EH-MAC

```
1:  $n_{est} \leftarrow 1$ (for ENAN);    $p_c \leftarrow 1.0$ (for AIMD)
2:  Wait for some random time
3:  Send a polling packet with $p_c = 1/n_{est}$ (for ENAN)
4:  Send a polling packet with $p_c$ (for AIMD)
5:  Listen to the channel
6:  if no sensor responds to the polling packet then
7:      $n_{est} \leftarrow \min(1, n_{est} - 1)$ (for ENAN)
8:      $p_c \leftarrow \min(p_c + p_{lin}, 1.0)$ (for AIMD)
9:  else if a data packet is successfully received then
10:     maintain value of $n_{est}$ (for ENAN) or $p_c$ (for AIMD)
11: else if packet loss due to poor channel conditions then
12:     maintain value of $n_{est}$ (for ENAN) or $p_c$ (for AIMD)
13: else if packet loss due to collision between two or more
    sender nodes then
14:     $n_{est} \leftarrow n_{est} + 1$ (for ENAN)
15:     $p_c \leftarrow p_c - (1 - p_{md})p_c$ (for AIMD)
16: end if
17: if end of active period reached then
18:     Go to charging state. Once enough energy is accumu-
        lated, repeat step 2
19: else
20:     Repeat step 2
21: end if
```

IV. PERFORMANCE EVALUATION

We use the Qualnet [11] network simulator to evaluate EH-MAC and other protocols in EH-WSNs using the specifications of the TI energy harvesting sensor node [12]. We deploy n EH-WSN nodes randomly over a simulation area measuring 500m by 500m. The TI node allows packet sizes larger than those using TinyOS, so each data packet (s_d) is set to 100 bytes to minimize overheads. Both polling (s_p) and acknowledgement (s_{ack}) packets are 15 bytes each. The buffer size is 10 data packets. The transmission rate (α) of the sensor node is 250 kbps, and the average transmission range is about 70m. We use a lognormal shadowing model and a Ricean fading model based on the radio characterization tests in [13].

The range of average energy harvesting rates (λ), from 2 mW to 20 mW, are obtained from datasheets of commercial energy harvesters and empirical measurements ([13],[14]). Since the node requires 72.6 mW (P_{rx}) to receive and 83.7 mW (P_{tx}) to transmit, it cannot be always active and the unpredictability in the energy harvesting process results in different charging times for each charge cycle. The charging time distribution is based on our empirical measurements [9].

The performance metrics are network capacity, throughput and fairness. The network capacity is computed by assuming that every node *always* has data packets to send. It is independent of the routing protocol or queue management scheme used, making it a useful and fair performance metric. The network capacity, C, is defined in bit-meters/second and is given by $C = (\sum_{i=1}^{n} \sum_{j=1}^{K_i} d_{i,j})/t$, where K_i is the number of packets sent successfully by node i, $d_{i,j}$ refers to the sender-receiver geographical distance for the j^{th} packet sent by node i and t is the simulation time.

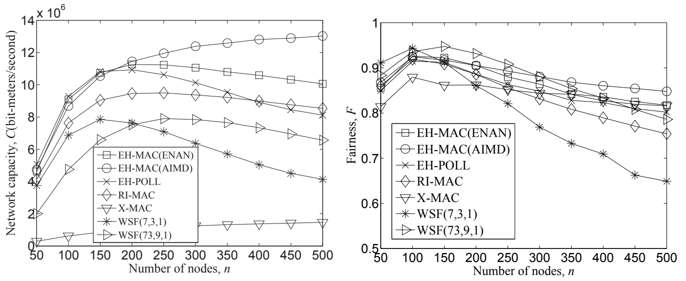
In event-driven WSN applications, data is sent to a sink (data collection point) whenever a node detects an event or anomaly. The main aim is to maximize the *rate of data packets* received by the sink. We deploy the sink at the center of the deployment area, and designate nodes furthest away from the sink as source nodes to demonstrate multi-hop capabilities, while the remaining nodes are relay nodes. We assume that the polling packet contains the location of the sender, so that simple geographic routing can be used to deliver data packets from the source nodes to the sink. Accordingly, a node will forward its data packet in response to a polling packet from any node that is nearer the sink than itself. The throughput, S, of an event-driven EH-WSN is $S = \sum_{i=1}^{n_s} H_i/t$, where H_i is the number of data packets received from source node i, n_s is the number of source nodes and t is the simulation time.

Fairness is defined using Jain's metric as $F = \frac{(\sum_{i=1}^{n} G_i)^2}{n(\sum_{i=1}^{n} G_i^2)}$ where G_i is the throughput of the i^{th} node or network capacity given to each node and n is the number of nodes. If each node has equal throughput/network capacity, F is 1. If only one node gets all the bandwidth, then $F \rightarrow 0$ as $n \rightarrow \infty$.

We compare EH-MAC with WSF [4], X-MAC [6] and RI-MAC [7], which are representative of the different types MAC protocols that can be used in EH-WSNs. There are two possible schedules for the WSF protocol, either (7,3,1) or (73,9,1) schedule. To determine the effectiveness of probabilistic polling, we have also included EH-MAC without any contention resolution scheme (EH-POLL). EH-POLL consists of all the features of EH-MAC except that p_c is set to 1.0 for every polling packet sent, therefore every node that receives the polling packet and has data packets to send will transmit a data packet. Each data point is derived from the average of 10 simulation runs of duration 100s each using different seeds.

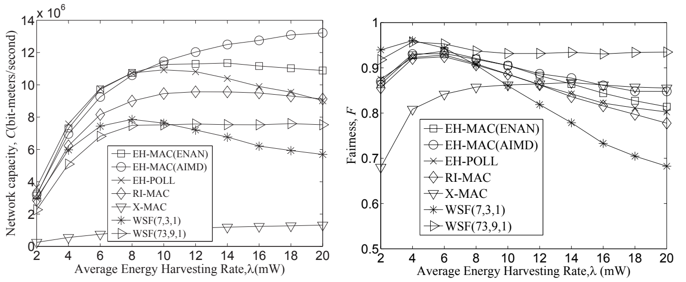
A. Network Capacity

We consider two different scenarios: In the first scenario, we vary the number of nodes, n from 50 to 500 using an energy harvesting rate, λ of 10 mW. In the second scenario, we vary λ from 2mW to 20mW with 200 nodes. Fig. 3 illustrates the network capacity and fairness results for varying node densities and energy harvesting rates. EH-MAC gives the highest network capacity because it aims to balance energy consumption with the amount of harvested energy. When the energy harvesting rates increase, more data can be transmitted. For the WSF protocol, the maximum duty cycle is 42.8% for the (7,3,1) block design and 12.3% for the (73,9,1) block design. Even if more energy is harvested, the extra energy cannot be utilized in WSF. Since X-MAC is designed for energy-constrained battery-operated WSNs, it typically operates at low duty cycles to achieve long lifetime and cannot make use of additional energy when energy harvesting rates increase.



(a) Network capacity for different node densities

(b) Fairness for different node densities



(c) Network capacity for different energy harvesting rates

(d) Fairness for different energy harvesting rates

Fig. 3. Network capacity for different MAC protocols

Another reason why EH-MAC outperforms WSF and X-MAC is that it incorporates a contention resolution scheme using probabilistic polling to reduce packet collisions. When node density or energy harvesting rate increases (decreases), the contention probability will decrease (increase) since there are more (fewer) active neighbors. For WSF or X-MAC, there is no mechanism to reduce packet collisions. At high node densities or energy harvesting rates, EH-MAC outperforms EH-POLL, demonstrating the effectiveness of probabilistic polling in reducing packet collisions.

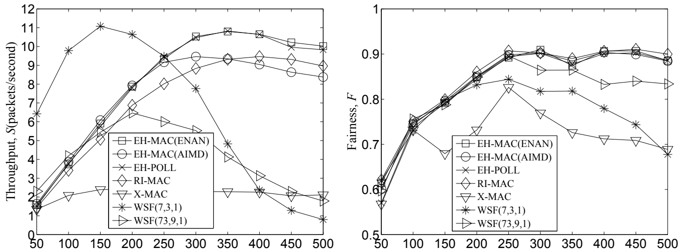
Although EH-MAC and RI-MAC are both receiver-initiated protocols, EH-MAC outperforms RI-MAC by up to 31% as EH-MAC can handle and recover from collisions faster than RI-MAC. In EH-MAC, nodes can send data packets as soon as they receive a polling packet without any delay but a backoff window is required in RI-MAC. EH-MAC also handles hidden terminal problems better than RI-MAC. In RI-MAC, the backoff window will only prevent collisions if neighboring nodes of the receiver can hear one another; this requirement is not needed in EH-MAC. Furthermore, the duration of every collision in EH-MAC is fixed at the duration of one packet transmission while the duration of a collision in RI-MAC may be much longer due to hidden terminals. EH-MAC can adapt to changing energy harvesting rates better than RI-MAC as the contention probability is adjusted after every polling packet while in RI-MAC, the backoff window size can only be changed after each backoff period of up to 255 slots.

EH-MAC(AIMD) outperforms EH-MAC(ENAN) at high node densities or energy harvesting rates because EH-MAC(AIMD) is more aggressive at reducing the contention probability and more conservative when increasing the contention probability. Since collisions take up more time and energy, EH-MAC(AIMD) outperforms EH-MAC(ENAN) as there are fewer collisions in EH-MAC(AIMD). The fairness metric refers to the network capacity given to each node. For all the data points, EH-MAC maintains high fairness (> 0.8) since every neighboring active node has equal probability of sending a data packet in response to a polling packet.

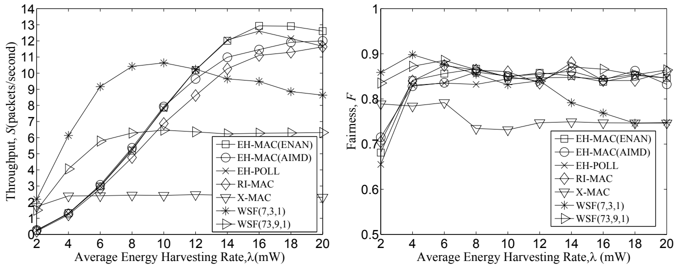
B. Event-driven WSN

The results are shown in Fig. 4 for different node densities and energy harvesting rates using 10 source nodes. WSF(7,3,1) gives higher throughput than EH-MAC at low node densities or energy harvesting rates but EH-MAC outperforms other MAC protocols at higher node densities or energy harvesting rates. This is unlike the case in the network capacity evaluation where EH-MAC outperforms all other protocols due to differences in the traffic model. At low node densities or energy harvesting rates, the WSF protocols work well because they achieve energy savings from the synchronization of time slots, thereby incurring less idle time. Furthermore, the probability of a collision (i.e., concurrent transmissions in the same time slot) is low. However, for higher node densities or energy harvesting rates, the probabilistic polling mechanism in EH-MAC reduces packet collisions and results in higher throughput. For X-MAC, the throughput is low as it is unable to adapt to different energy harvesting rates because it has fixed duty cycles. Similarly, EH-MAC outperforms RI-MAC for the same reasons as in the network capacity evaluation. EH-MAC is able to give high fairness because probabilistic polling ensures that all nodes have equal opportunities to transmit or receive, therefore the sink can receive data from all the source nodes.

The difference in throughput between EH-MAC and EH-POLL is marginal because there are only 10 source nodes (i.e., low traffic conditions). However, when we increase the traffic by designating 10% of the total number of nodes as sources, we observe, from Fig. 5, then EH-MAC gives higher throughput than RI-MAC and EH-POLL (up to 27% and 37% respectively), demonstrating that probabilistic polling is an effective contention resolution scheme.



(a) Throughput for different node densities (b) Fairness for different node densities



(c) Throughput with different energy harvesting rates (d) Fairness with different energy harvesting rates

Fig. 4. Performance evaluation for different MAC protocols with 10 source nodes for event-driven WSNs using different node densities and energy harvesting rates

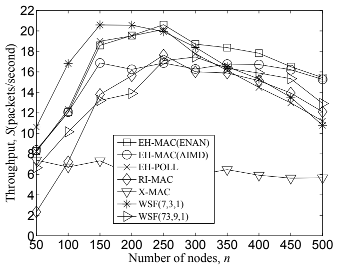
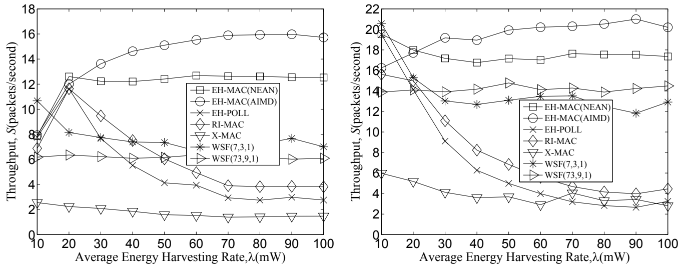


Fig. 5. Throughput with varying number of source nodes ($n_s = n/10$)

Next, we illustrate the performance results for a wider range of energy harvesting rates. Fig. 6 shows the throughput values for energy harvesting rates between 10 mW to 100 mW using 200 sensor nodes with varying number of source nodes. At 100 mW, the sensor nodes can always be active with very high probabilities since it exceeds the operating power requirements. The results show that EH-MAC is able to give high throughput even for higher energy harvesting rates by adjusting the contention probability dynamically. For WSF and X-MAC, the duty cycle is fixed, therefore any additional harvested energy has minimal impact on throughput. EH-MAC outperforms RI-MAC due to a better contention resolution scheme using probabilistic polling.

V. CONCLUSION

Using Energy Harvesting Wireless Sensor Networks (EH-WSNs) is very attractive as it can eliminate the problem of replacing batteries. However, many networking protocols for WSNs often trade throughput and latency for a decrease in energy consumption to extend network lifetime. Since nodes in EH-WSNs can replenish their energy, new network protocols that can match energy consumption with the energy harvesting rate are needed. This paper describes EH-MAC, a novel MAC protocol designed for multi-hop EH-WSNs. EH-MAC comprises a probabilistic polling mechanism to reduce packet collisions. The contention probability and the sending frequency of polling packets are dynamically adjusted according to changing energy harvesting rates, node densities and traffic load to reduce overheads and interference. EH-MAC also reduces the hidden terminal problem in multi-hop scenarios. Extensive simulation results show that EH-MAC can achieve high throughput and fairness compared to other MAC protocols for EH-WSNs.



(a) Event-driven WSN with 10 source nodes (b) Event-driven WSN with 20 source nodes

Fig. 6. Throughput of different MAC protocols for varying energy harvesting rates from 10 mW to 100 mW using 200 sensor nodes

REFERENCES

[1] J. A. Paradiso and T. Starner, "Energy Scavenging for Mobile and Wireless Electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.

[2] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," *IEEE Communications Surveys and Tutorials*, 2010.

[3] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[4] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous Wakeup for Ad Hoc Networks," in *MobiHoc*, 2003, pp. 35–45.

[5] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *SenSys*, 2004, pp. 95–107.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *SenSys*, 2006, pp. 307–320.

[7] 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," in *SenSys*, 2008, pp. 1–14.

[8] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *ACM Mobicom*, 1999, pp. 174–185.

[9] Z. A. Eu, H.-P. Tan, and W. K. G. Seah, "Design and performance analysis of MAC Schemes for Wireless Sensor Networks Powered by Ambient Energy Harvesting," *Ad Hoc Networks*, vol. 9, no. 3, pp. 300–323, 2011.

[10] Z. A. Eu, P. Lee, and H.-P. Tan, "Classification of Packet Transmission Outcomes in Wireless Sensor Networks," in *IEEE ICC*, 2011.

[11] "Qualnet network simulator." [Online]. Available: http://www.scalable-networks.com

[12] "MSP430 Solar Energy Harvesting Development Tool (eZ430-RF2500-SEH) from Texas Instruments." [Online]. Available: http://www.ti.com

[13] Z. A. Eu, H. P. Tan, and W. K. G. Seah, "Wireless Sensor Networks Powered by Ambient Energy Harvesting: An Empirical Characterization," in *IEEE ICC*, 2010.

[14] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-Aware Energy Synchronization for Wireless Sensor Networks," in *MobiSys*, 2009, pp. 319–332.