A Study of MAC Schemes for Wireless Sensor Networks Powered by Ambient Energy Harvesting

Zhi Ang Eu
NUS Graduate School for Integrative Sciences and
Engineering, National University of Singapore
CeLS, #05-01, 28 Medical Drive, Singapore 117456

g0601792@nus.edu.sg

Winston K. G. Seah, Hwee-Pink Tan
Networking Protocols Department,
Institute for Infocomm Research (I^{2}R), A*STAR
1 Fusionopolis Way, #21-01 Connexis, Singapore 138632

{winston, hptan}@i2r.a-star.edu.sg

ABSTRACT
Energy consumption is a perennial issue in the design of wireless sensor networks which typically rely on portable sources like batteries for power. Recent advances in ambient energy harvesting technology have made it a potential alternative source of energy for powering wireless sensor networks. In this paper, we study the performance of four different medium access control (MAC) protocols based on CSMA and polling techniques for wireless sensor networks which are powered by ambient energy harvesting. First, we define our model for describing the ambient energy harvesting process. Next, we derive the main performance metrics which are the per-node throughput of each sensor node (R) and the network throughput (S) which is the rate of sensor data received by the sink. We validate the analytical models using simulations, and the results show that neither CSMA-based nor polling protocols always gives the best performance results. These results aim to provide insights to the design of algorithms and protocols for wireless sensor networks that rely on ambient energy harvesting for power which is different from the typical assumption of limited power as in the case of batteries.

Categories and Subject Descriptors
C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms
Algorithms, Performance, Design

Keywords
Wireless Sensor Networks, Medium Access Control, CSMA, Polling, Energy Harvesting

1. INTRODUCTION
Much research on sensor networks have focused on extending the lifetime of sensor networks which are assumed to rely on finite energy sources like batteries for power. In contrast, wireless sensor networks (WSNs) powered by ambient energy harvesting (which we refer to as WSN-HEAP in this paper) are more useful and economical in the long-term as they can operate for very long

periods of time until hardware failure because ambient energy may be harvested from the environment at all times. Examples of ambient energy sources include light, vibration, heat and wind. Therefore, WSN-HEAP present promising solutions for solving the energy constraints of WSNs. However, as the rate of charging is usually much lower than the rate of energy consumption for the sensor nodes, WSN-HEAP nodes can only be awake for a short period of time before it needs to shut down in order to recharge. Moreover, the time taken to charge up the sensor is not constant due to environmental factors. Figure 1 shows the salient difference in the characteristics of these two energy sources. Our main contribution is the performance study of four different fundamental MAC protocols when used in WSN-HEAP. Our analysis focuses on the throughput of each sensor node R and the network throughput S, which is the rate at which the sink receives data from all the sensor nodes. We validate our analytical models by comparing the numerical predictions with simulation results.
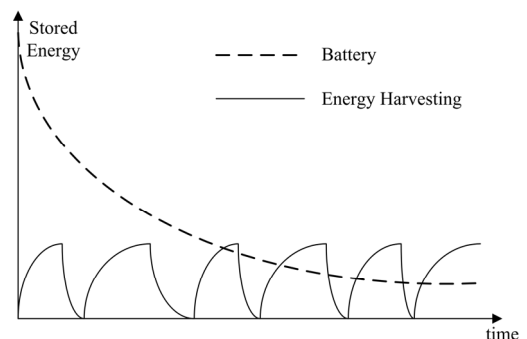

Figure 1. Characteristics of Energy Sources.

2. RELATED WORK
Most sensor nodes used in WSNs today rely on a limited energy source like batteries to operate. One attempt [1] to solve the energy problem is to make use of some mobile sensor nodes to deliver energy to other sensor nodes. Another solution that has been adopted is to make use of sensor nodes that rely on energy harvesting devices [2-4] for power. Since batteries have limited recharge cycles, supercapacitors with unlimited recharge cycles are an attractive option for use in such WSNs to replace batteries because they can operate perpetually without the need for replacement. Some examples of WSN-HEAP have been deployed in testbeds. For example, in [5], 557 solar-powered motes have been used to evaluate robust multi-target tracking algorithms. Other solar-powered sensor network testbeds are illustrated in [6] and [7]. There are also commercially available sensor nodes which rely on ambient energy harvesting for power. The devices developed by Microstrain [8] harvest and use energy from two sources, viz. solar and mechanical energy. To date, none of these

efforts address issues related to the networking aspects of WSNs. Instead, the focus is on the efficiency and viability of the energy harvesting method. Furthermore, most of the reported work focused on harvesting energy to supplement battery power while we focus on using the harvested energy as the only energy source.

Various MAC protocols have been designed for WSNs. In [9], sleep and wakeup schedules are proposed to reduce energy usage and prolong network lifetime at the expense of longer delays. Since these schemes assume the use of batteries in their scenarios, energy conservation therefore is a key consideration. However, in our scenario, ambient energy is harvested which makes the optimal use of this ambient energy to maximize throughput and minimize delays our key considerations. Sift [10] is another protocol designed for event-driven sensor networks to minimize collisions in a slotted CSMA system. Our application domain is different as the sensor node would send sensor data to the sink whenever it accumulates enough energy, making Sift unsuitable for use in our scenario. In this paper, we evaluate the performance of different fundamental CSMA and polling MAC protocols when used in WSN-HEAP as a precursor to developing new medium access control schemes for such sensor networks.

3. WSN-HEAP ARCHITECTURE

WSN-HEAP are very useful in applications where sensors cannot be easily accessible after deployment, and the replacement of the exhausted on-board power source is not feasible. Furthermore, power sources like batteries are inappropriate in some applications due to environmental concerns and the risk of battery leakage. Examples of such applications include sensors for structural health monitoring [11-13] where sensors are embedded into buildings and structures. In [13], a network architecture consisting of one sink with many WSN-HEAP nodes is proposed. This type of architecture is the focus of this paper. We consider a network scenario consisting of n WSN-HEAP nodes and one sink in a single-hop environment. The sink is a data collection point which is connected to power mains, and therefore does not need to be charged. All the sensors can transmit data to the sink directly. All sensors and the sink can hear one another, and there are no hidden terminals. The notations used in our analytical model are summarized in Table 1.

4. CSMA FOR WSN-HEAP

4.1 Slotted CSMA Protocol Description

We first consider a modified version of a slotted CSMA protocol which is used in IEEE 802.11 [16] and 802.15.4 [17] networks. In the slotted CSMA model, there are three states in which a sensor could be in as illustrated by the state transition diagram in Figure 2. They are the *charging*, *carrier sensing* and *transmit* states.
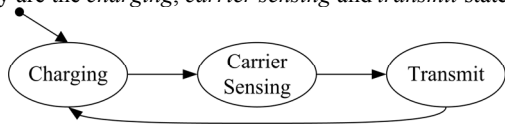

Figure 2. State transition diagram for slotted CSMA.

In the slotted form of the CSMA protocol, each slot consists of the time to transmit one data packet denoted by t_{tx} and the hardware turnaround time denoted by t_{ta}. We let the duration of each slot be t_s. A sensor would only transmit its data packet when the current transmission in the slot has ended. If there is no transmission in the current slot by any sensor, the sink would transmit a synchronization packet in that slot. To simplify our analysis, we set the size of the synchronization packet such that the end of transmission time of the synchronization packet coincides with the end of a data transmission if there is a data transmission in that slot. The data transmission timings are illustrated in Figure 3 which shows that the 1st, 2nd and 4th transmission slots are data sent by the sensors while the sink would transmit a synchronization packet in the 3rd and 5th slots once it detects no sensor has transmitted in that slot. The time taken to determine whether the channel is idle or not when it transits into the carrier sensing state is denoted by t_{cca}.

Table 1. Notations used in the analytical models

|	
E_{rx}	Energy required to receive a data packet
E_{ta}	Energy required to change state (from receive to transmit or from transmit to receive)
E_{tx}	Energy required to send a data packet
E_f	Energy of a fully charged sensor node
G	Random variable denoting the energy harvesting rate
n	Number of sensor nodes in the network
P_{rx}	Power needed when the sensor is in receive state
P_{ta}	Power needed to switch from receive to transmit or from transmit to receive
P_{tx}	Power needed when the sensor is in transmit state
R	Per-node throughput of each sensor
S	Network throughput
s_{ack}	Size of an acknowledgment packet from the sink
s_d	Size of a data packet
s_p	Size of a polling packet
t_{ack}	Time taken to send an acknowledgement packet
t_{cca}	Time taken to determine whether there is an ongoing data transmission by detecting whether the channel is clear or not
t_f	Average time taken to charge up the sensor if the initial energy of the sensor is 0
t_{poll}	Time to send a polling packet
t_s	Time of a transmission slot in the slotted CSMA model
t_{tx}	Time to send a data packet
t_{ta}	Hardware turnaround time from receive state to transmit state or from transmit state to receive state
α	Transmission rate of the sensor
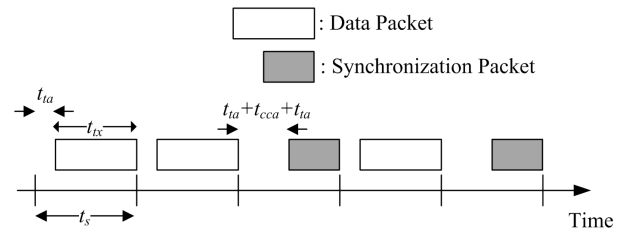
Figure 3. Transmission timings in slotted CSMA.

A cycle starts when the sensor goes into the charging state and ends when it leaves the transmit state. The energy harvesting rate is not constant because it depends on environmental factors such as the placement of sensor and weather conditions. Therefore, we model the charging process by a continuous random variable G which denotes the energy harvesting rate, and the time in charging state by the continuous random variable A. The waiting time until the start of the next slot is modeled by the continuous random variable B as shown in Figure 4.
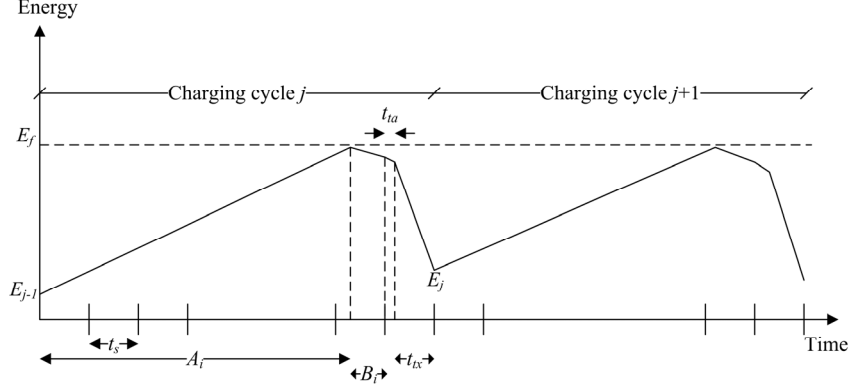
Figure 4. Energy model for slotted CSMA.

We do not assume a linear charging process in our analysis; this is depicted in Figure 4 only for the purpose of illustration. When the stored energy of the sensor reaches E_f, it *wakes up* and goes into the carrier sensing state to wait for the start of the next time slot. At the beginning of the next time slot, it will go into the transmit state and start sending its sensed data to the sink. We denote the energy required to receive and transmit a data packet by E_{rx} and E_{tx} respectively. If α is the transmission rate of the sensor, then the time taken to transmit one data packet of size s_d bytes (i.e., time in transmit state) is $t_{tx} = 8s_d / \alpha$.

If P_{rx} and P_{tx} are the receive and transmit power of the sensor respectively, then we have

$$E_{rx} = 8s_d P_{rx} / \alpha ,$$
$$E_{tx} = 8s_d P_{tx} / \alpha .$$

Similarly, if E_{ta} and P_{ta} are the energy and power required during hardware turnaround time respectively, then $E_{ta} = P_{ta}t_{ta}$.

4.2 Throughput Analysis of Slotted CSMA

To analyze the performance of the CSMA protocol, we need to find the probability of a collision when a sensor transmits a data packet. We let A_i and B_i be the time in charging state and carrier sensing state (waiting time till start of next slot) of sensor node i respectively and let $t_s=t_{ta}+t_{tx}$ be the slot duration. In any given time slot, we have

P(node i transmits data packet successfully)
   =P(only node i transmits)
   =P(node i transmits) P(no other node transmits)

$$= \frac{t_s}{A_i + B_i + t_s} \prod_{k \neq i}^{n} \frac{A_k + B_k}{A_k + B_k + t_s} ,$$

since the charging processes of all the sensor nodes are independent. We let R_i be the node throughput of sensor node i and S be the network throughput (expected number of packets received by sink per second). Then, we have the following:

$$R_i = \frac{1}{t_s} E\left[\frac{t_s}{A_i + B_i + t_s} \right] \prod_{k \neq i}^{n} E\left[\frac{A_k + B_k}{A_k + B_k + t_s} \right]$$

$$\approx \frac{1}{E[A_i] + E[B_i] + t_s} \prod_{k \neq i}^{n} \frac{E[A_k] + E[B_k]}{E[A_k] + E[B_k] + t_s} ,$$

$$S = \sum_{i=1}^{n} R_i .$$

If the charging processes of all sensor nodes are statistically identical, then the per-node throughput R is given by

$$R = \frac{1}{E[A] + E[B] + t_s} \frac{(E[A] + E[B])^{n-1}}{(E[A] + E[B] + t_s)^{n-1}}$$
$$= \frac{(E[A] + E[B])^{n-1}}{(E[A] + E[B] + t_s)^{n}} \tag{1}$$

and the network throughput, S, is given by

$$S = \frac{n(E[A] + E[B])^{n-1}}{(E[A] + E[B] + t_s)^{n}} . \tag{2}$$

The derivations of $E[A]$ and $E[B]$ are given in the Appendix.

Next, we compute the network capacity which is defined as the maximum value of S. From (2), we know that S first increases and then decreases for increasing values of n. Therefore, there is an optimal value of n that maximizes S. We denote this optimal value of n by n^* and the network capacity by S^*. To determine n^*, we solve

$$\frac{dS}{dn} = 0 . \tag{3}$$

By substituting (2) into (3), we get

$$n^* = \frac{1}{\ln(E[A] + E[B] + t_s) - \ln(E[A] + E[B])} , \tag{4}$$

$$S^* = \frac{n^*(E[A] + E[B])^{n^*-1}}{(E[A] + E[B] + t_s)^{n^*}} . \tag{5}$$

Special Case: Exponentially Distributed Charging Time
A more accurate model can be obtained if A is exponentially distributed and the values of B, t_{ta} and t_{tx} are much smaller than A. In this model, the interval between successive wakeups for one sensor is also approximately exponentially distributed with rate $1/(E[A]+E[B]+t_s)$ where the values of $E[A]$ and $E[B]$ are given in the Appendix by (12) and (13) respectively. Since there are n sensors in the network, the interval between successive wakeups for all the sensors is exponentially distributed with rate $n/(E[A]+E[B]+t_s)$. This means that the number of wakeups in any time interval can be modeled as a Poisson process. There would be a successful data transmission in the next time slot if there is only one wakeup in the current time slot. We let X be the number of wakeups in a transmission time slot which is of duration t_s. The probability of a successful transmission is

$$P(X=1) = e^{-\frac{nt_s}{E[A]+E[B]+t_s}} \times \frac{nt_s}{E[A]+E[B]+t_s} \ .$$

Since there are $1/t_s$ timeslots per second,

$$S = \frac{1}{t_s}P(X=1) = \frac{ne^{-nt_s/(E[A]+E[B]+t_s)}}{E[A]+E[B]+t_s} \ , \qquad (6)$$

$$R = \frac{S}{n} = \frac{e^{-nt_s/(E[A]+E[B]+t_s)}}{E[A]+E[B]+t_s} \ . \qquad (7)$$

By substituting (6) into (3), we get

$$n^* = (E[A]+E[B]+t_s)/t_s \ , \qquad (8)$$

$$S^* = e^{-1}/t_s \ . \qquad (9)$$

4.3 Validation of analytical model

We need to validate the analytical model using simulation results to ensure that our model is reasonably accurate despite the approximations made. We use the Qualnet [14] network simulator to validate our models. In our simulations, we model a sensor network application which uses sensor nodes to monitor structures in construction sites. Energy is harvested from the vibration due to machines and ground movement. In [2], it was reported that vibrational microgenerators can provide energy at a rate of 4 µW/cm³ due to human motion and 800 µW/cm³ due to machines. Therefore, we assume that the average energy harvesting rate is 1.5 mW after considering the size of the sensors in use today. We have also referred to the specifications of MICAz sensor mote [15] manufactured by Crossbow Technology Inc. to compute the transmit and receive powers. We consider the power consumption of two components of the sensor node which are the processor and the RF transceiver as these two components account for most of the power needed for the mote. For the MICAz sensor mote, the current draw for the processor is 8 mA. The current draw for the RF transceiver is 19.7 mA and 17.4 mA for receiving and transmitting respectively at maximum transmit power. The current draw during hardware turnaround time is estimated to be the average of the current draw during the receive and transmit states. The assigned parameter values are shown in Table 2.

Table 2. Values of various parameters used in simulation

Parameter	Value
n	ranges from 1 to 500
P_{rx}	83.1 mW
P_{ta}	79.65 mW
P_{tx}	76.2 mW
s_{ack}	10 bytes
s_d	100 bytes
s_p	10 bytes
t_{cca}	0.128 ms
t_{tx}	4.096 ms
t_{ta}	0.192 ms
$E[G]$	1.5 mW for fixed t_f
α	250 kbps
Simulation Time	500 seconds

We consider two different distributions for the charging time. For uniform distribution, we assume that the full charging time is uniformly distributed in $[t_f - t_f/2, t_f + t_f/2]$ where t_f is the average time taken to fully charge a sensor node if it is initially uncharged. The value of t_f can be computed as:

$$t_f = \frac{E_f}{E[G]} \ .$$

For exponential distribution, we assume that the full charging time is exponentially distributed with a mean of t_f seconds. First, we fixed the value of t_f and vary the number of nodes, n. The results for uniformly distributed and exponentially distributed charging times are illustrated in Figure 5. We can observe that the network throughput is almost invariant with the distribution of the charging time. Furthermore, we find that analytical values given by (2) and (6) do not differ too much. This implies that (2), which is applicable to any distribution in general, is a good approximation to (6) which assumes exponentially distributed charging times. Therefore, due to space constraints, we would only present the graphs for exponentially distributed charging times for the rest of the protocols in the paper.
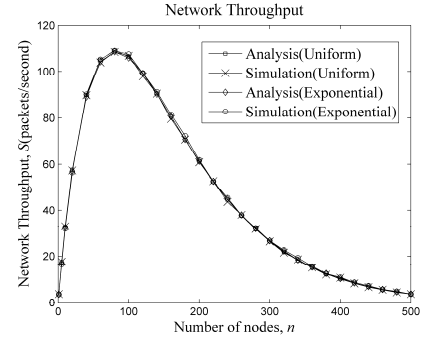

Figure 5. Network throughput for varying values of n with slotted CSMA.

As the average charging time may not be fixed in real scenarios because it is dependent on environmental factors, we need to ensure that our model is accurate for different charging rates. Therefore, we fixed the number of nodes, n at 200 and vary t_f from 0.25 to 2.5 seconds. The results are illustrated in Figure 6 which shows that our numerical results from the analytical model closely match the simulation results, thus validating our analysis.
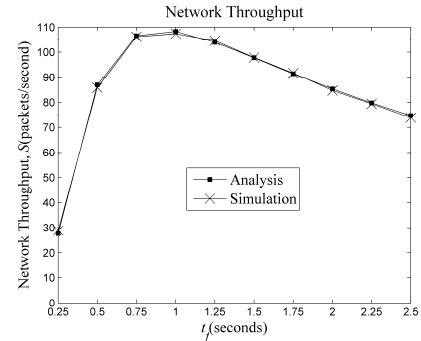

Figure 6. Network throughput for varying values of t_f for slotted CSMA (exponentially distributed charging times).

4.4 Unslotted CSMA Protocol Description

The performance of the slotted CSMA protocol can be improved by considering two main drawbacks of the protocol. Firstly, for large values of n, the number of collisions can be reduced by having a backoff scheme. Secondly, by not having time slots, energy required is reduced during the carrier sensing state. This is because once the node senses that the channel is busy, it can go

into the charging state to recharge immediately. In this section, we present an unslotted form of the CSMA protocol. There are five states in which a sensor could be in as illustrated by the state transition diagram in Figure 7. They are the *charging*, *carrier sensing*, *receive*, *idle* and *transmit* states.
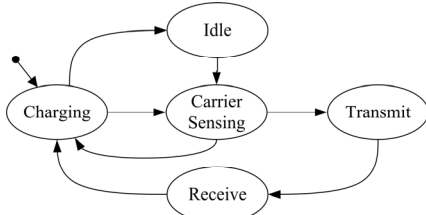

Figure 7. State transition diagram for unslotted CSMA.

Initially, the sensor is uncharged so it would be in the charging state. When the supercapacitor is full, it would go into the carrier sensing state to determine whether the channel is free. If the channel is free, it would transmit the data packet. Then, it would move into the receive state to wait for an acknowledgment (ACK) packet from the sink. After receiving the ACK packet from the sink, it would return to the charging state. Figure 8 illustrates the energy model for a successful data transmission if the channel is free at the first sensing attempt.
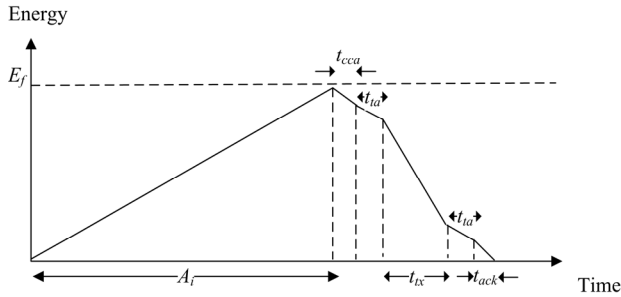

Figure 8. Energy model of a successful transmission in unslotted CSMA.

If the channel is busy, it would perform a backoff and go back into the charging state. If the supercapacitor is full but the sensor has not reached the end of its backoff period, then it would be in the idle state until the end of the backoff period when it would go into the carrier sensing state. The energy model when backoffs are needed is shown in Figure 12. The average backoff period is doubled under two situations as shown in the flowchart in Figure 9. The first situation is when it senses that the channel is not free. The second situation is when it does not receive an ACK from the sink after transmitting a data packet. The average backoff time is doubled after every backoff attempt by doubling the backoff exponent (*BE*). Each backoff duration ranges from one unit backoff period to a maximum of 255 unit backoff periods. Each unit backoff period is 320 microseconds which is the duration of a time slot specified in IEEE 802.15.4 standards [17]. In each backoff period, the node would be recharged until the maximum energy is obtained.
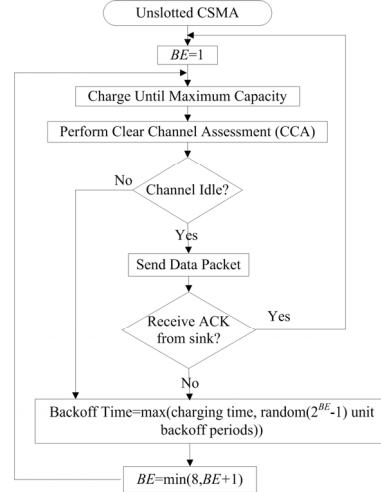

Figure 9. Flowchart illustrating backoff mechanism for unslotted CSMA.

We implement the unslotted form of CSMA protocol in the network simulator and the results are shown in Figure 10. We observe that unslotted CSMA provides higher network throughput. However, there is a drawback compared to the slotted CSMA protocol. Since the unslotted CSMA protocol incorporates a backoff mechanism, some nodes may get higher throughput than other nodes in a short time period. To illustrate this, we set the simulation time to be 60 seconds and obtain the minimum, average and maximum number of packets received by the sink per second from each sensor as shown in Figure 11. It can be observed that the variation in per-node throughput for unslotted CSMA is larger than the variation in the slotted CSMA protocol.
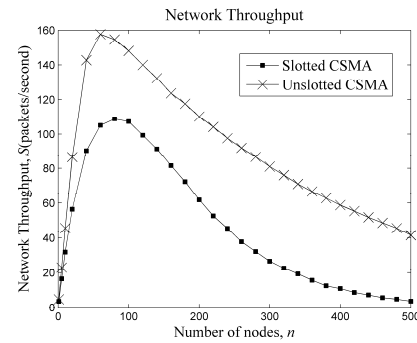

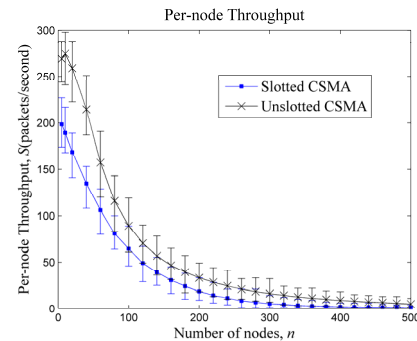**Figure 10. Network throughput obtained with slotted and unslotted CSMA.**

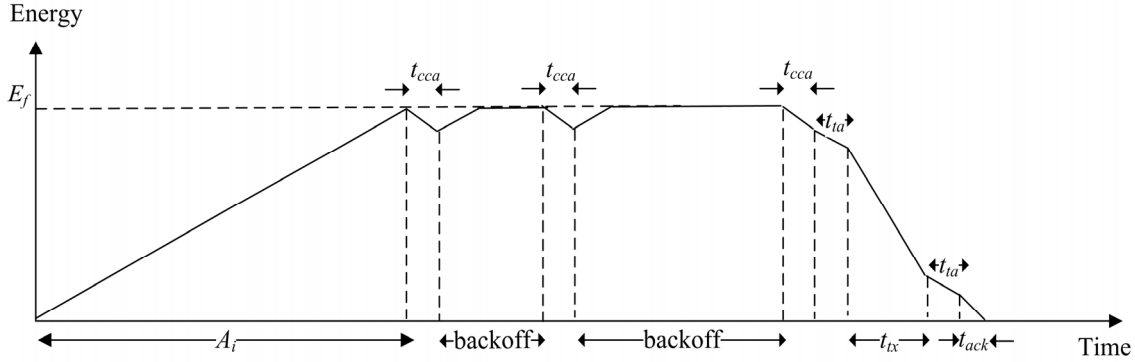**Figure 11. Per-node throughput obtained with slotted and unslotted CSMA.**

Figure 12. Energy model for unslotted CSMA when backoff periods are required.

5. POLLING FOR WSN-HEAP

5.1 Description of ID Polling

With an identity (ID) polling protocol, the sink will transmit a polling packet to request for data from a sensor. Each polling packet will contain the sensor's unique ID which is randomly chosen from the set of all sensor nodes. If the sensor being polled is in the charging state, then the sink will not be able to get a response from the sensor. If the sensor being polled is in receive state, it will send its sensed data to the sink after it receives the polling packet. The state transition diagram as shown in Figure 13 is similar to that of the unslotted CSMA protocol. However, there is a new possible transition from the receive state to the charging state since the sensor has to recharge if its ID does not match the ID value in the polling packet it receives in the receive period.
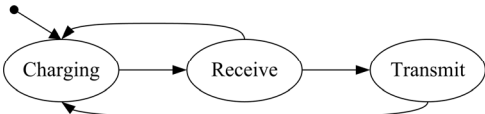

Figure 13. State transition diagram for ID Polling.

Each polling packet is separated from a data packet by t_{ta} which is the time required for the sink to change states. For an unsuccessful poll, there is a minimum separation of $2t_{ta}+t_{cca}$ between two successive polling packets which is the time required to determine whether there is any response from the sensor before another polling packet is sent as illustrated in Figure 14.
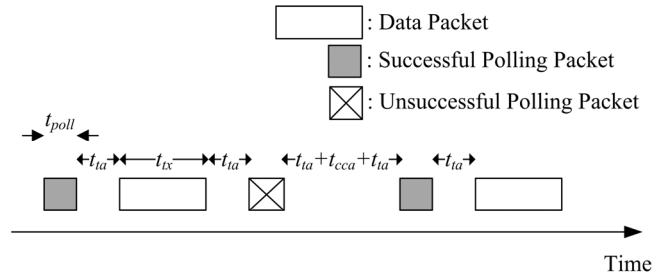

Figure 14. Transmission timings for polling.

5.2 Analytical model for polling

We let C be a continuous random variable to denote the time to recharge the sensor and D be a continuous random variable to denote the time before a transmission if the sensor is being polled. If the sensor is not being polled by the sink and its energy level falls below the energy required to transmit one packet, the sensor will need to harvest additional energy until the total energy reaches E_f. The energy model is illustrated in Figure 15.
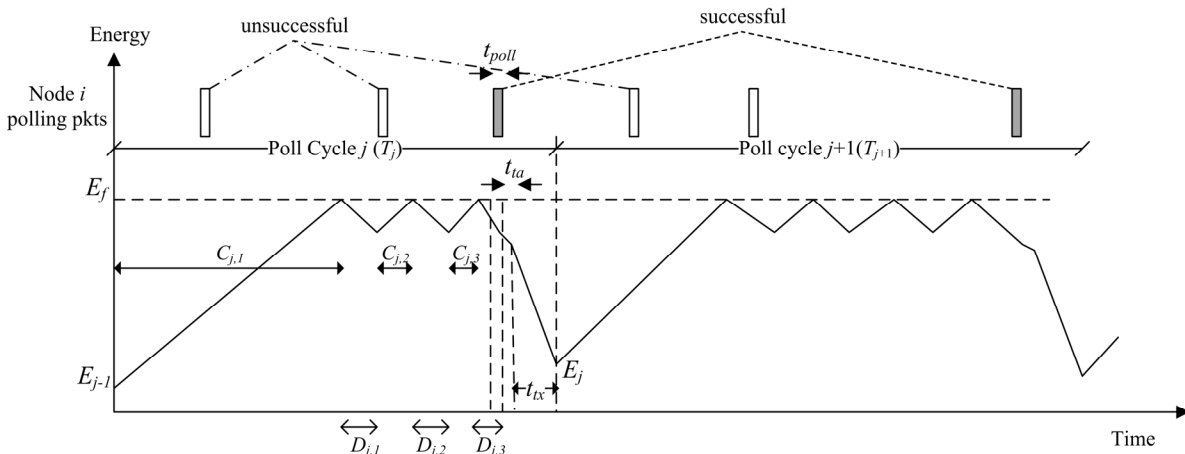

Figure 15. Energy Model for ID Polling.

5.3 Throughput Analysis for ID Polling

We define a cycle to be the time taken to send a data packet from a sensor to the sink successfully. A cycle starts immediately when the sensor transits to the charging state after the end of a successful transmission. The cycle ends at the end of a successful data transmission to the sink. During the cycle, the sensor may transit into the charging state several times as it waits for its turn to be polled by the sink. We let the probability that the sensor is in receive state when the sink polls the sensor for data be p.

Since t_{poll} is the time taken to transmit a polling packet of size s_p bytes, it is given by

$$t_{poll} = 8s_p / \alpha .$$

For a successful poll, the time required is $t_{poll}+2t_{ta}+t_{tx}$, where t_{tx} is the transmission time of a data packet. For an unsuccessful poll, the time required is $t_{poll}+2t_{ta}+t_{cca}$. Therefore, each sensor requires an average time of $t_{poll}+2t_{ta}+pt_{tx}+(1-p)t_{cca}$ to poll. Since the sensors are polled randomly, the average total time to poll each sensor once is $n[t_{poll}+2t_{ta}+pt_{tx}+(1-p)t_{cca}]$ and the average number of data packets received from all the sensors is np. Since a cycle is also the time to receive an average of one data packet from each sensor or a total n data packets, the average cycle time is $n[t_{poll}+2t_{ta}+pt_{tx}+(1-p)t_{cca}]/p$. Therefore,

$$R \approx \frac{1}{\text{average cycle time}} = \frac{p}{n[t_{poll} + 2t_{ta} + pt_{tx} + (1-p)t_{cca}]} , \quad (10)$$

$$S = nR = \frac{p}{t_{poll} + 2t_{ta} + pt_{tx} + (1-p)t_{cca}} . \quad (11)$$

The derivations of p and the network capacity S^* are given in the Appendix.

5.4 Validation of analytical model

We validated our analytical model of ID Polling using simulations and the results are shown in Figure 16.
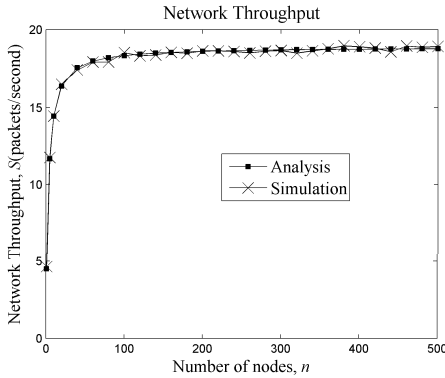

Figure 16. Network throughput for varying values of n with ID Polling (Exponentially distributed charging times).

Next, we fixed the number of nodes, n at 200 and vary t_f from 0.25 to 2.5 seconds. From the results shown in Figure 17, we observe that our model is sufficiently accurate as shown by the close matching of the simulation values with the analytical predictions.
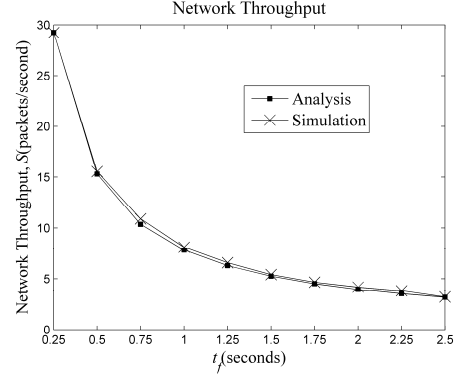

Figure 17. Network throughput for varying values of t_f with ID Polling (Exponentially distributed charging times).

5.5 Probabilistic Polling

From Figure 10 and Figure 16, we observe that ID polling performs poorly compared to the CSMA protocols in WSN-HEAP. This is because the probability of a successful poll is very low, since the time in which a sensor spends in receive state is much shorter than the time in charging state. Furthermore, we have to know the unique IDs of all the sensors in the network which may not be possible in a totally distributed environment. We may even waste time polling a failed sensor. Therefore, we propose an improvement by using probabilistic polling.

In probabilistic polling, instead of having the sensor's unique ID in the polling packet, we have a number, p_c, in the polling packet that indicates the probability that a sensor should transmit its data packet. Upon receiving the polling packet, a node would generate a random number $x \in [0, 1]$. The sensor transmits its data packet if $x < p_c$; otherwise, it will either remain in the receive state or transit to the charging state when its energy is depleted.

The value of p_c will therefore determine the performance of the protocol. We use an additive-increase multiplicative-decrease (AIMD) protocol to adjust the value of p_c dynamically so that ideally only one out of all the sensors that are in the receive state when polled would transmit a data packet. The algorithm is described below:

Algorithm to determine p_c
1. Set $p_c = 0.01$.
2. Send a polling packet.
3. If no sensor responds to the polling packet,
   Set $p_c = \min(p_c+0.01, 1.0)$.
4. If a data packet is successfully received from one of the sensors, keep p_c at current value.
5. If there is a collision between two or more sensors as indicated by a corrupted data packet,
   Set $p_c = p_c /2$.
6. Repeat step 2.

We implement probabilistic polling in our network simulator and compare it with ID polling. The results shown in Figure 18 validate the effectiveness of our approach as it clearly shows that probabilistic polling achieves much higher network throughput than ID polling.
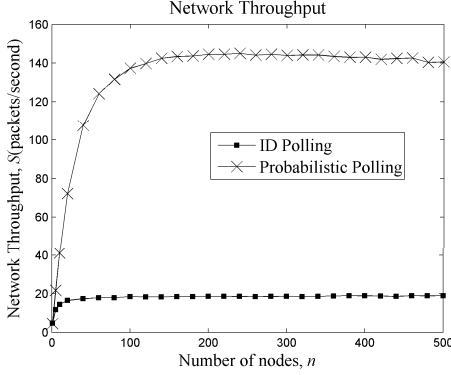
Figure 18. Comparison of ID and Probabilistic Polling.

We now compare all the four MAC protocols presented in this paper. The results are shown in Figure 19 for exponentially distributed charging times. From the graphs, we can conclude that there are two important values of n. The maximum network throughput is obtained with unslotted CSMA when there are n_1 sensor nodes. For values of n below n_2, unslotted CSMA outperforms all other protocols while the probabilistic polling protocol gives the best throughput for $n > n_2$.
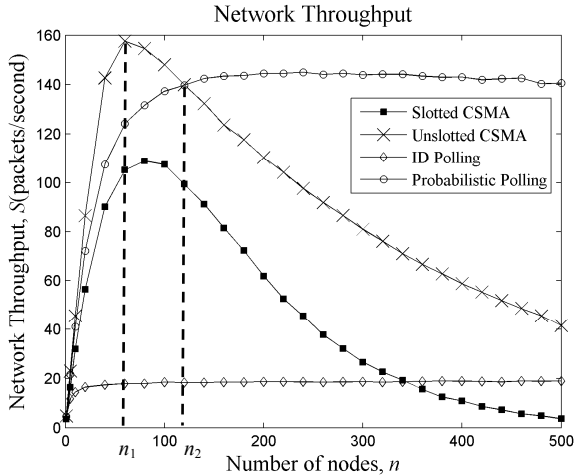

Figure 19. Comparison of CSMA and polling MAC protocols.

6. CONCLUSION

In this paper, we studied four different fundamental MAC protocols that can be used in wireless sensor networks powered by ambient energy harvesting. We presented analytical models for the slotted CSMA and ID polling MAC protocols by deriving the performance metrics, sensor and network throughput, as functions of the number of sensor nodes, charging rate, transmission time, transmit power and receive power. This gives us insights on how the performance metrics are affected by different parameters. We have also obtained the network capacity of slotted CSMA and ID polling which is useful in QoS provisioning. Our analytical models were validated using simulations developed on the QualNet Simulator [14]. In addition, we studied the performance of unslotted CSMA and probabilistic polling protocols which improve throughput as compared to slotted CSMA and ID polling respectively. Our performance analysis shows that the unslotted CSMA protocol gives higher throughput until the point when the number of sensor nodes exceeds a certain threshold after which

the probabilistic polling protocol gives the highest throughput among the schemes studied. The results from this study provide insights for our future work involving the design of algorithms and protocols for WSN-HEAP which are suitable for applications like structural health monitoring of critical infrastructures and in-situ long-term environmental monitoring.

Appendix

Derivation of $E[A]$ and $E[B]$ for slotted CSMA:

We define E_f to be the maximum energy required to ensure that the node can transmit at the start of a time slot by listening to the end of a transmission of a previous data packet by another node or a synchronization packet by the sink. It can be calculated by

$$E_f = (t_{cca} + t_{ta})P_{rx} + E_{rx} + E_{ta} + E_{tx}.$$

Referring to Figure 4, we can write the following for charging cycle j:

$$E_j = E_{j-1} + G_j A_i - (B_i P_{rx} + E_{ta} + E_{tx}) + G_j(t_s + B_i),$$

where G_j is the charging rate for cycle j.

Taking expectations, and under steady-state conditions (where $E[E_j]=E[E_{j-1}]$), we have:

$$E[A] = \frac{(E[B]P_{rx} + E_{ta} + E_{tx}) - E[G](E[B] + t_s)}{E[G]}. \quad (12)$$

The minimum value of B_i is t_{cca} when the node senses the channel just before the end of a transmission slot and the maximum value is $(t_{cca}+t_s)$ when the node senses the channel after another transmission has ended. Since B_i is uniformly distributed over the interval $[t_{cca}, t_{cca}+t_s]$, we get

$$E[B] = 0.5t_s + t_{cca}. \quad (13)$$

Derivation of p for ID polling:

We define E_f to be the energy required to receive at least one polling packet and transmit one data packet. It can be computed by considering the case in which a node wakes up just after the beginning of a transmission of a polling packet. Therefore,

$$E_f = (2t_{poll} + 2t_{ta} + t_{tx})P_{rx} + E_{ta} + E_{tx}.$$

We let the kth polling cycle of during T_k be the duration between the $(k-1)^{th}$ and k^{th} successful packet transmission from node i as shown in Figure 15. Each T_k begins with residual energy E_{k-1} and comprises charging periods of total duration $C_k = \sum_m C_{k,m}$ and receive periods of total duration $D_k = \sum_m D_{k,m}$.

Since a poll is successful only if node i is in receive state, we can approximate p as follows:

$$p \approx \frac{E[D]}{E[C]+E[D]} \times \frac{\max(D_{k,m}) - t_{poll}}{\max(D_{k,m})}. \quad (14)$$

The constant factor in (14) is to reduce p since the sink has to transmit at least t_{poll} before the end of the node's receive period as illustrated in Figure 20.
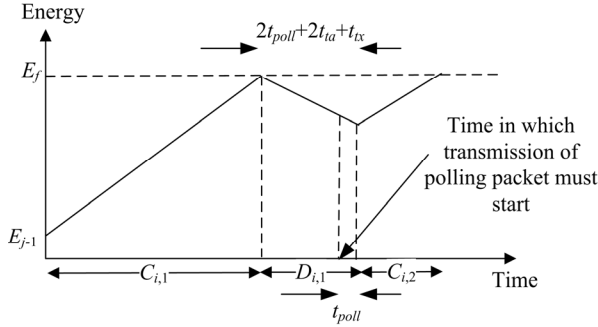
Figure 20. Timings of ID Polling.

Since we define the maximum receive period to be the time to receive at least one polling packet, $\max(D_{k,m})$ is $(2t_{poll}+2t_{ta}+t_{tx})$. Therefore, we have

$$p \approx \frac{E[D]}{E[C]+E[D]} \times \frac{t_{poll}+2t_{ta}+t_{tx}}{2t_{poll}+2t_{ta}+t_{tx}}. \qquad (15)$$

Next, we consider the energy flow of polling cycle k:

$$E_k = E_{k-1} + C_k G_k - D_k P_{rx} - E_{ta} - E_{tx} + (D_k + t_{ta} + t_{tx})G_k .$$

Taking expectations, and under steady-state conditions (where $E[E_k]=E[E_{k-1}]$), we have

$$E[C]E[G] = E[D]P_{rx} + E_{ta} + E_{tx} - (E[D]+t_{ta}+t_{tx})E[G] . \qquad (16)$$

In addition, since $E[T]=E[C]+E[D]+t_{tx}+t_{ta}$, we have

$$\frac{n[t_{poll}+2t_{ta}+pt_{tx}+(1-p)t_{cca}]}{p} = E[C]+E[D]+t_{tx}+t_{ta} . \qquad (17)$$

By solving (15), (16) and (17), we can get the values of $E[C]$, $E[D]$ and p.

Derivation of S* for ID polling:

We let the maximum throughput using the polling protocol be S^*. When n increases, p increases, and hence S increases. However, since the rate of increase in p decreases as n increases, to find S^*, we use

$$S^* = \lim_{n \to \infty} S .$$

When n increases, the time used to harvest energy for listening to polling packets increases. When $n \to \infty$, most of the energy harvested is consumed in the receive state. Furthermore, since $(t_{ta}+t_{tx}) << E[D]$ when n is large, (16) can be approximated using

$$E[C] \approx \frac{E[D]P_{rx} - E[D]E[G]}{E[G]} . \qquad (18)$$

By substituting (18) into (15), we get

$$p^* = \frac{E[G]}{P_{rx}} \times \frac{t_{poll}+2t_{ta}+t_{tx}}{2t_{poll}+2t_{ta}+t_{tx}} . \qquad (19)$$

By substituting (19) into (11), we can get S^*.

7. REFERENCES

[1] Rahimi, M., Shah, H., Sukhatme, G. S., Heidemann, J., and Estrin, D., "Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network", in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Sep 2003, pp. 19-24, Taipei, Taiwan.

[2] Paradiso, J. A. and Starner, T., "Energy Scavenging for Mobile and Wireless Electronics", in IEEE Pervasive Computing, Vol. 4, No. 1, 2005, pp.18-27.

[3] Raghunathan, V., Ganeriwal, S., and Srivastava, M., "Emerging Techniques for Long Lived Wireless Sensor Networks", in IEEE Communications Magazine, Vol. 44, No. 4, April 2006, pp. 108-114.

[4] Simjee, F. I. and Chou, P. H., "Efficient Charging of Supercapacitors for Extended Lifetime of Wireless Sensor Nodes", in IEEE Transactions on Power Electronics, Vol. 23, No. 3, May 2008, pp. 1526-1536.

[5] Dutta, P., Hui, J., Jeong, J., Kim, S., Sharp, C., Taneja, J., Tolle, G., Whitehouse, K., and Culler, D., "Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments", in Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN), Apr 2006, pp. 407-415, Nashville, TN, USA.

[6] Sikka, P., Corke, P., Valencia, P., Crossman, C., Swain, D., and Bishop-Hurley, G., "Wireless Adhoc Sensor and Actuator Networks on the Farm", in Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN), Apr 2006, pp. 492-499, Nashville, TN, USA.

[7] Corke, P., Valencia, P., Sikka, P., Wark, T., and Overs, L., "Long-Duration Solar-powered Wireless Sensor Networks", in Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets), Jun 2007, pp. 33-37, Cork, Ireland.

[8] Microstrain, http://www.microstrain.com.

[9] Ye, W., Heidemann, J., and Estrin., D., "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks", in IEEE/ACM Transactions on Networking, Vol. 12, No. 3, June 2004, pp. 493-506.

[10] Tay, Y. C., Jamieson, K., and Balakrishnan, H., "Collision-Minimizing CSMA and Its Applications to Wireless Sensor Networks", in IEEE Journal on Selected Areas in Communications, Vol. 22, No. 6, Aug 2004, pp. 1048-1057.

[11] Wu, W.-J., Chen, Y.-F., Chen, Y.-Y., Wang, C.-S., and Chen, Y.-H., "Smart Wireless Sensor Network Powered by Random Ambient Vibrations", in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Oct 2006, pp. 2701-2708, Taipei, Taiwan.

[12] Kottapalli, V. A., Kiremidjian, A. S., Lynch, J. P., Carryer, E., Kenny, T. W., Law, K. H., and Lei, Y., "Two-tiered wireless sensor network architecture for structural health monitoring", in Proceedings of the SPIE's 10th Annual International Symposium on Smart Structures and Materials, Mar 2003, San Diego, CA, USA.

[13] Park, G., Farrar, C. R., Todd, M. D., Hodgkiss, W., and Rosing, T., "Energy Harvesting for Structural Health Monitoring Sensor Networks", Technical Report, Los Alamos National Laboratory, February 2007.

[14] Qualnet Network Simulator 4.5, http://www.scalable-networks.com.

[15] MICAz sensor mode specifications, available from http://www.xbow.com.

[16] IEEE 802.11 Stds, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2003.

[17] IEEE 802.15.4-2006 Stds, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).