

Performance Modeling and Simulation Studies of MAC Protocols in Sensor Network Performance

Herman Sahota

Ratnesh Kumar

Ahmed Kamal

Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011

Abstract—The use of wireless sensor networks is essential for implementation of information and control technologies in precision agriculture. We present our design of network stack for such an application where sensor nodes periodically collect data from fixed locations in a field. Our design of the physical (PHY) layer consists of multiple power modes in both the receive and transmit operations for the purpose of achieving energy savings. In addition, MAC layer is designed which uses these multiple power modes to save energy during the wake-up synchronization phase. We also present analytical models and simulation studies to compare the energy consumption of our MAC protocol with that of the popular S-MAC protocol and show that our protocol has better energy efficiency as well as latency in a periodic data collection application.

I. INTRODUCTION

Precision agriculture refers to the use of information and control technologies in agriculture. In [1], we presented the design of our sensor network for automated collection of soil data from a farm-field. The application requires the collection of soil data over the entire duration of crop season(s) at a sampling frequency of about once an hour and a spatial resolution of about 50×50 square-meters (see Fig. 1). Sensor nodes are placed in a rectangular grid at the required spatial resolution and are integrated into a network that collects data periodically. The data is relayed to a sink node located at one of the corners of the rectangular field at regular time intervals, consistent with the measurement period.

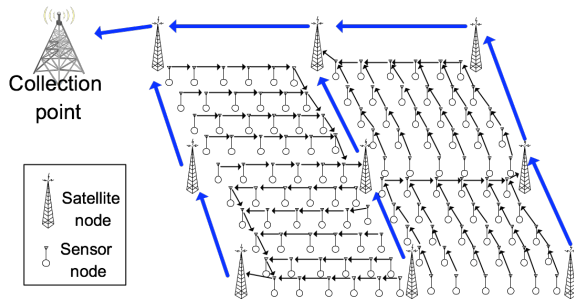


Fig. 1: Wireless sensor network, showing sensor and satellite station deployments.

Communication between sensor nodes in a field and the satellite nodes at the corners of the field is asymmetric, in the

This work was supported in part by the National Science Foundation under the grants CNS-0626822, NSF-ECS-0601570, NSF-ECCS-0801763, NSF-CCF-081141, and NSF-ECCS-0926029.

sense that a transmission from a satellite node can reach all sensor nodes in the field, but a transmission from a sensor node reaches adjacent sensor nodes only, and can reach a satellite node if it is close enough (see Fig. 1). The data collection occurs periodically in *rounds* where satellite nodes take turns acting as sink nodes and collect information from the sensor nodes in a field to relay the collected information to the basestation where the information is processed. Additionally, satellite nodes also dispatch routing and scheduling information to the nodes in the field and participate in the sensor localization process. Given a sink node for a round, a routing tree rooted at that node is constructed for achieving load-balancing (a node will forward its data to that neighbor which has witnessed least depletion of energy so far). See Fig. 2. The scheduling is done complying with the order imposed by the routing.

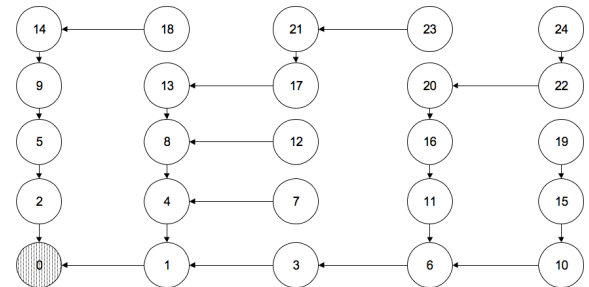


Fig. 2: Routing tree for 5×5 grid used for simulation.

A key concern for sensor networks is its energy efficiency. To address this concern, we proposed in [1] a new MAC protocol that uses new radio modes for performing wake-up synchronization, and that too at a group-level (a receiver and all its upstream senders are synchronized simultaneously). In this paper we study the performance of our protocol through mathematical modeling and validate it with simulations.

The following are the contributions of this paper:

- We present an analytical approach to model the performance (throughput, latency and energy consumption) of the MAC protocol (ours and the standard S-MAC [2]) under a given routing strategy.
- We validate our analytical models by developing simulations in nesC and Python using event driven framework provided by TOSSIM.

II. ENERGY EFFICIENT WAKE-UP AND DATA COLLECTION

All nodes receive common routing and scheduling data at the beginning of each round from the sink node of that round and use this event to synchronize their clocks. However the clocks drift during the course of the round owing to the fact that sensor nodes are implemented using inexpensive crystals that can drift of the order of 40 parts per million. When two nodes are ready to synchronize, their clocks could drift by an amount $\pm\Delta$ relative to a true clock. Therefore in the worst case, the receiver of a wake-up signal needs to wake up for a time that is on the order of 4Δ [1] in order to make sure it can capture the wake-up synchronization signal from its transmitter. Therefore, the energy consumed by the receiver can be significant if Δ is large, which is the case for long data-collection rounds.

In our application, nodes are synchronized at the beginning of each data-collection round, and during this period the node clocks drift, requiring synchronization between sender and receiver nodes. Our strategy is aimed at reducing the energy consumption during the wake-up synchronization phase of the transmission. In the current designs of MAC protocols ([3], [2], [4]), the wake-up synchronization between a transmitter and a receiver is established using the normal transmission-power and receiver-sensitivity levels. We consider a modified wake-up synchronization scheme in which a node saves its battery energy consumption by reducing its own receiver sensitivity level. In contrast, we increase the transmission-power level of the wake-up synchronization signal to allow for a successful detection by a lower sensitivity level receiver. It turns out that this new scheme saves the overall energy consumption during the synchronization phase since the protocol uses a relatively short pulse for wake-up signal [1]. Additionally, we take advantage of the convergecast nature of sensor network traffic to minimize the number of high energy pings transmitted. This is achieved by making the downstream node wake up all its upstream neighbors with a common ping signal. Existing sensor network MAC protocols such as S-MAC [2], T-MAC [4] and WiseMAC [3] establish individual link communications independently, regardless of whether two or more links share a common node. Besides the savings at the MAC level, our routing level strategy aims at balancing the energy consumption among the sensor nodes in the entire field.

We refer to the mode of short-duration and high-power wake-up signal transmission as the *ping* mode. Likewise, the mode of operation of the receiver circuit in the degraded sensitivity is termed as *drowsy*. These two modes of the radio are used in the wake up strategy employed by our MAC protocol to save on energy for wake-up synchronization. We call our MAC protocol PD-MAC (letters ‘P’ and ‘D’ stand for ping and drowsy, respectively).

III. MATHEMATICAL ANALYSIS AND SIMULATION

We establish the performance improvements in latency and energy achieved by PD-MAC by comparing its performance with that of the most commonly used S-MAC protocol, in the

setting of our application, using analytical models as well as simulations. For comparison purposes, both protocols operate under the same routing and scheduling information made available by the network layer. For the simulation studies we consider a 25 node network placed in a 5x5 rectangular grid, and the routing tree as depicted in Fig. 2. A corner node, node 0 acts as the sink at which all the data from the network is collected. The scheduling strategy ensures that no two links in the network are active at the same time to avoid interference and also ensures that a node begins transmitting data to its downstream node only after it has already attempted to collect data from all of its upstream nodes.

The differences between S-MAC and PD-MAC lie in the implementation of the routes and schedules and also the synchronization protocol. In S-MAC, the synchronization handshake as well as the data/ack exchange occurs in a pairwise manner between one sender node and one receiver node. If a node has more than one upstream neighbors, then the respective sender→receiver links are scheduled in a sequence, and for each pair a maximum of N_s synchronization attempts is allowed, and once the synchronization is established, a total of N_d data attempts is allowed.

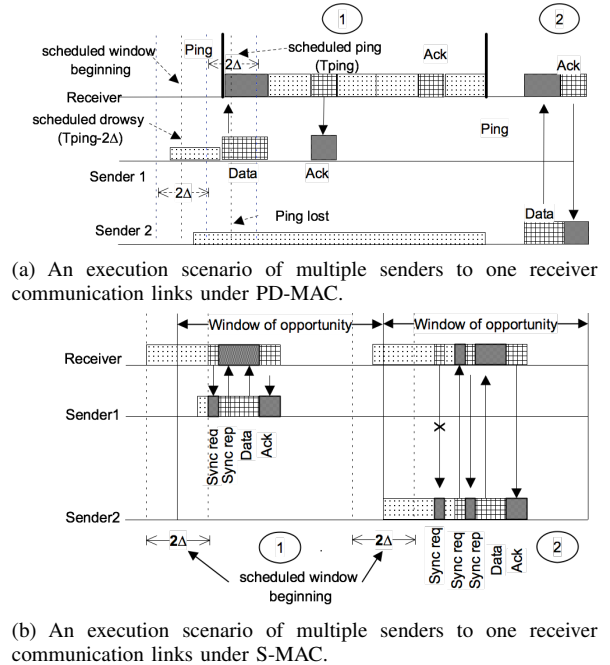


Fig. 3

In S-MAC, when a node wakes up to establish a communication link with its neighbor, it waits for a predefined duration before transmitting a synchronization request packet so as to guarantee that the other node on the link is awake to receive the synchronization request. The duration of this period is accordingly chosen to be the worst case relative drift (2Δ) between the clocks of the two nodes. Accordingly, letting $2T_s$ denote the time to send synchronization request and receive reply packet, the *discovery duration* (T_{DD}), which is defined as the period of time a node must use when it

wakes up to discover any existing neighboring sleep-listen schedules, lasts $T_{DD} = 2\Delta + 2T_S$ units of time. The node that receives the synchronization request packet replies with a synchronization reply packet. Keeping the primary goal of comparing latency and energy savings, we make a simplifying assumption that the reply packet is delivered error free. In addition, in regular S-MAC, the synchronization reply (a.k.a rebroadcast of the synchronization packet) by the node that receives a synchronization request happens after a random back-off to prevent multiple nodes from transmitting at the same time. However, in our application setting, only a sender-receiver pair can be active at a time. Hence, a back-off is not required. The two nodes keep awake (i.e., maintain a 100% duty cycle) until they expend all synchronization and data transfer attempts or until data transmission succeeds. This way S-MAC is not penalized with a longer round duration by being forced to sleep part of the time due the choice of a less than 100% duty cycle. Also, a sender node is guaranteed to have data to send to its downstream neighbor when it wakes up eliminating the need for a sleep-listen schedule. Fig. 3 shows a possible execution scenario of a schedule consisting of one receiver and two sender nodes by S-MAC, under the choice of parameters $N_s = 2$ and $N_d = 1$.

On the other hand, in PD-MAC, a node synchronizes all of its upstream nodes with a common ping. A maximum number (N_s) of ping attempts are allowed in case of unsuccessful synchronization. Between successive pings, the upstream neighbors are allowed a maximum of N_d data transmission attempts in a time division manner. This process of synchronization and data collection constitutes the communication time of a node in PD-MAC. A possible execution of PD-MAC under the same scenario as S-MAC is also shown in Fig. 3.

Next, we present our analytical models as well as simulation results to compare the performance (throughput, latency and energy consumption) of two MAC protocols. Note throughput is determined by the average number of data packets collected per unit time, and can be obtained by taking the ratio of the average data-count at the sink node per round and the average latency of the round. So it suffices to determine data-count and latency per round.

A. Modeling data-count at sink

We model the probability distribution of number of data readings collected at each node at the end of each round. $P_D^m(i)$ denotes the probability that node m gathers i data readings (including locally generated reading) after it has attempted to collect data from its upstream neighbors, U_m . We evaluate $P_D^m(i)$ recursively as shown in Equation (1). The leaf nodes of the routing tree do not have any upstream neighbors and so they collect exactly one data unit. Therefore, the base case of the recursion is: For all leaf nodes m , $P_D^m(i) = 1$ for $i = 1$ and $P_D^m(i) = 0$ for $i > 1$. For the non-leaf nodes, $P_D^m(i)$ is given by the following formula:

$$P_D^m(i) = \sum_{\substack{S_m \subseteq U_m: \\ \sum_{k \in S_m} i_k = i-1}} \left\{ \left(\prod_{k \in S_m} P_D^k(i_k) P_{\text{suc}}(i_k) \right) \cdot \left(\prod_{k \in U_m \setminus S_m} P_D^k(i_k) P_{\text{fail}}(i_k) \right) \right\} \quad (1)$$

The above expression computes the probability that a subset of upstream neighbors, $S_m \subseteq U_m$, collectively transmits $i-1$ data units to node m , where the upstream neighbor $k \in S_m$ transmits i_k data units such that $\sum_{k \in S_m} i_k = i-1$. $P_{\text{suc}}(i_k)$ denotes the probability that synchronization is established in at most N_s attempts and a data packet containing i_k data units is transmitted successfully in at most N_d attempts, whereas $P_{\text{fail}}(i_k) = 1 - P_{\text{suc}}(i_k)$.

$$P_{\text{suc}}(i_k) = \sum_{s=1}^{N_s} (1-q) q^{s-1} \sum_{r=1}^{N_d} \left((1-p_e)^{D(i_k)} (1 - (1-p_e)^{D(i_k)})^{r-1} \right), \quad (2)$$

where q is probability of synchronization failure, p_e is the bit error rate, $D(i_k)$ denotes the bit length of a data packet containing i_k data units. The expression for the data-count probability for PD-MAC and S-MAC remains the same as given in Equations (1) and (2) except for the value of the q parameter. For S-MAC, this is computed in accordance with the bit error probability and the number of bits in the synchronization packet while for PD-MAC, this denotes the probability of failure of ping, which is a much lower value owing to its shorter length and no data content.

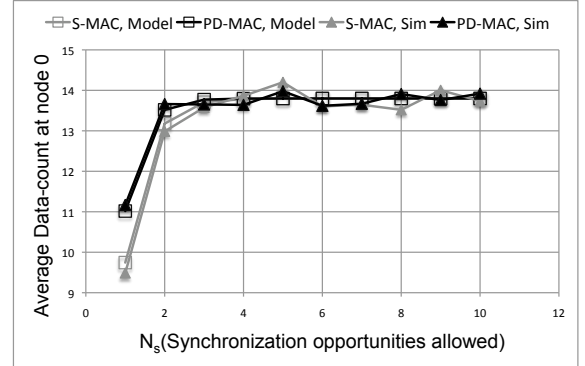


Fig. 4: Expected data count of a round at sink.

We used a bit error rate of 0.01, ping transmission error probability of 0.1, 8-bits per data unit, 8-header bits per packet, 8-bits payload per S-MAC synchronization request/reply packet, ping pulse duration of 0.1s, and a baud-rate of 1200bps was used for our performance evaluation studies. The expected data-count at the sink-node 0 is given by $\sum_{i=1}^N i P_D^0(i)$, where N is the total number of nodes in the field. Fig. 4 shows the expected data count at node 0 versus the number of synchronization attempts allowed for the two protocols. Both protocols have comparable performance

as far as the data-count at the sink node is concerned. This behavior is expected because we allow the same number of synchronization and data transmission attempts for the two MAC protocols.

B. Modeling latency or round duration

We define the *communication time* of a node as the time taken by a node to synchronize with and collect data from all its upstream senders. Round duration is the sum of the communication times for all nodes in the routing tree. In the following two subsections we model the communication time for a node for the two MAC protocols.

1) *Modeling communication time for S-MAC*: Let T_{km} denote the expected communication time for a receiver node m , corresponding to an upstream neighbor k as modeled in Equation (3). Here $P_{\text{sync}}(i)$ denotes the probability that synchronization succeeds in i th attempt ($i \leq N_s$) and $P_{\text{sync-fail}}$ denote the probability that synchronization fails all N_s attempts. If synchronization fails N_s attempts, data phase does not take place. However, if synchronization succeeds in at most N_s attempts, the two nodes attempt to exchange the data packet for up to a maximum of N_d attempts.

$$T_{km} = \sum_{i=1}^{N_s} P_{\text{sync}}(i) \cdot [T_{\text{sync}}(i) + T_{\text{data}}^k] + P_{\text{sync-fail}} \cdot T_{\text{no-sync}}, \quad (3)$$

where $T_{\text{sync}}(i)$ denotes the expected duration of the synchronization phase given that synchronization succeeds in i th attempt, T_{data}^k denotes the expected data phase duration consisting of a maximum of N_d attempts (its value depends on k as different upstream neighbors have different number of data units to send), and $T_{\text{no-sync}}$ denotes the expected duration of the synchronization phase when synchronization fails all N_s attempts. Equation (4) computes $T_{\text{sync}}(i)$, whereas the formulae for $T_{\text{no-sync}}$ and T_{data}^k are given in Equations (5) and (6) respectively.

$$T_{\text{sync}}(i) = \left\lfloor \frac{i+1}{2} \right\rfloor T_{DD} + \{(i+1) \bmod 2\} \mathbb{E}(Y) \quad (4)$$

Equation (4) can be understood as follows: the two nodes that are attempting to synchronize with each other take turns in sending synchronization request packets to each other. Therefore, if synchronization occurs in i th attempt, the node that starts first takes a total of $\lfloor \frac{i+1}{2} \rfloor$ turns, each of which lasts for the discovery duration of T_{DD} . In addition, if the synchronization succeeds in an even-numbered attempt, an additional delay corresponding to the difference in the wake-up times of the two nodes, denoted Y in Equation (4), is also incurred denoting synchronization request transmission by node that wakes up second. For example, Fig. 5 shows $T_{\text{sync}}(5)$ and Y . Similarly,

$$T_{\text{no-sync}} = \left\lfloor \frac{N_s+1}{2} \right\rfloor T_{DD} + \{(N_s+1) \bmod 2\} \mathbb{E}(Y) \quad (5)$$

Note, $Y = |X_k - X_m|$, where the random variables for wake-up time of k and m , $X_k, X_m \sim U(0, 2\Delta)$. Then the

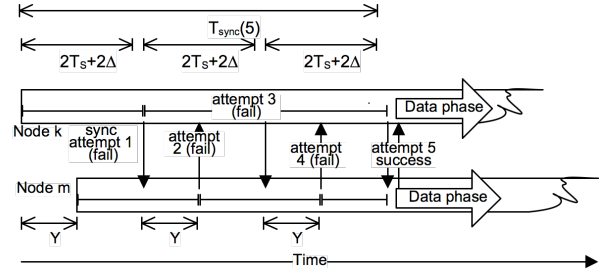


Fig. 5: Synchronization phase in S-MAC. Y represents the absolute difference between the wake up times of the sender and receiver nodes.

expected value of Y is given by: $\mathbb{E}(Y) = \frac{2}{3}\Delta$.

We now compute T_{data}^k , the expected time to transmit data by an upstream neighbor k in a total of N_d attempts. Let R_k denote the sub-routing-tree rooted at node k . Then, since this subtree has size $|R_k|$ and since each node senses one data unit, node k can have at most $|R_k|$ data units to transmit to its downstream node m (since some data packets may be lost). The probability distribution of the number of data packets contained at every node is expressed by Equation (1). Let $T_D(l)$ denote the time-slot for l data units and $T_A(s)$ denote the time-slot for an acknowledgement vector for s number of senders (for S-MAC $s = 1$, always). Note $T_D(l) = D(l)/BPS$ and $T_A(s) = A(s)/BPS$, where BPS denotes the bit rate of the radio transceiver and $A(s)$ denotes the bit length of an acknowledgement packet containing a bit vector of length s . Then the size of time-slot needed to accommodate the maximum-sized data from node k (of size $|R_k|$) and one acknowledgement to node k is given by $T_{DA}^k = T_D(|R_k|) + T_A(1)$. We can model the expected data phase duration as follows:

$$T_{\text{data}}^k = \sum_{l=1}^{|R_k|} P_D^k(l) \left(\sum_{r=1}^{N_d} P_{\text{suc}}^r(l) \cdot r \cdot T_{DA}^k + P_{\text{fail}}^{N_d}(l) \cdot N_d \cdot T_{DA}^k \right), \quad (6)$$

where $P_D^k(l)$ is defined in Equation (1), $P_{\text{suc}}^r(l)$ is the probability of successful transmission of a data packet containing l data units in the r th attempt. Since the packet error rate for a data packet containing l data units is given by $p_l = 1 - (1 - p_e)^{D(l)}$, the formulae for $P_{\text{suc}}^r(l)$ and $P_{\text{fail}}^{N_d}(l)$ are as follows:

$$P_{\text{suc}}^r(l) = (1 - p_l)p_l^{r-1}; \quad P_{\text{fail}}^{N_d}(l) = p_l^{N_d}. \quad (7)$$

2) *Modeling communication time for PD-MAC*: In PD-MAC, node m attempts to synchronize with all of its upstream neighbors U_m using a common ping. Here, we obtain a recursive model for the communication time of a node m .

Let $T^m(i, r, S_m, D_m, Q_m)$ denote the remaining communication time, given that $i - 1$ synchronization attempts have taken place and $r - 1$ data attempts associated with the i th synchronization attempt have taken place. $S_m \subseteq U_m$ denotes the subset of upstream neighbors that are synchro-

$$T(i, r, S_m, D_m, Q_m) = \begin{cases} 0, & \text{if } C := [(D_m = U_m) \vee (i = N_s + 1)] \\ T_{SYN} + \sum_{R_m \subseteq U_m \setminus S_m} \sum_{C_m \subseteq R_m} \hat{T}_{DA}^m + q^{|U_m \setminus S_m \setminus R_m|} (1 - q)^{|R_m|} P_{\text{suc}}(C_m) P_{\text{fail}}(R_m \setminus C_m) \\ T(i + \lfloor (r/N_d) \rfloor, (r \bmod N_d) + 1, S_m \cup R_m, D_m \cup C_m, R_m), & \text{if } \neg C, r = 1 \\ \sum_{C_m \subseteq Q_m \setminus D_m} \hat{T}_{DA}^m + P_{\text{suc}}(C_m) P_{\text{fail}}(Q_m \setminus D_m \setminus C_m) \\ T(i + \lfloor (r/N_d) \rfloor, (r \bmod N_d) + 1, S_m, D_m \cup C_m, Q_m), & \text{if } \neg C, r > 1, \end{cases} \quad (8)$$

nized so far, $D_m \subseteq S_m$ denotes the subset of upstream nodes that have successfully delivered their data so far, and $Q_m \subseteq U_m \setminus S_m$ denotes the subset of upstream nodes that have synchronized in the most recent (i^{th}) synchronization attempt. Q_m gets updated only following a synchronization attempt (and remains unaltered following a data transmission attempt). A recursive model for $T(i, r, S_m, D_m, Q_m)$ is given by Equation (8), where T_{SYN} denotes the duration of ping, and $\hat{T}_{DA}^m = T_A(|U_m|) + \sum_{k \in U_m} T_D(|R_k|)$ represents the duration of entire time-slot accommodating data packets by all senders and acknowledgement packet by the downstream node m . Then, $T(1, 1, \emptyset, \emptyset, \emptyset)$ gives the total communication time for node m .

Case 1 in Equation (8) denotes the base condition. The remaining communication time $T(S_m, D_m, Q_m, i, r)$ is zero if node m has already received data from all its upstream neighbors (captured by the condition: $D_m = U_m$), or if it has already expended all N_s synchronization attempts along with the N_d data attempts associated with each synchronization attempt (captured by the condition: $i = N_s + 1$). Case 2 models the i^{th} ping transmission which takes a duration of T_{SYN} and the ensuing (first) data attempt which takes a duration of \hat{T}_{DA}^m . In a synchronization attempt ($r = 1$) a subset $R_m \subseteq U_m \setminus S_m$ may get synchronized (captured in the first summation) with a probability of $q^{|U_m \setminus S_m \setminus R_m|} (1 - q)^{|R_m|}$ and participate in at most N_d data attempts. During the first data attempt a subset $C_m \subseteq R_m$ of nodes may be successful in data transmission (captured in the second summation) with a probability of $P_{\text{suc}}(C_m) P_{\text{fail}}(R_m \setminus C_m)$. Case 3 represents the r^{th} data attempt for $r > 1$, which takes the duration \hat{T}_{DA}^m . During a data attempt a subset $C_m \subseteq Q_m \setminus D_m$ of newly synchronized nodes may be successful in transmitting data with a probability of $P_{\text{suc}}(C_m) P_{\text{fail}}(Q_m \setminus D_m \setminus C_m)$. (Note only the newly synchronized nodes attempt to send data, for the nodes previously synchronized already attempted a maximum of N_d allowed attempts.)

The results for the round-duration of the model as well as the simulations for both S-MAC and PD-MAC are plotted against the number of synchronization attempts in Fig. 6. It should be noted that PD-MAC achieves a 25% shorter round duration than S-MAC. This is because PD-MAC combines the wake-up synchronization of all nodes that are upstream neighbors of a node. Such combination is not allowed in S-MAC.

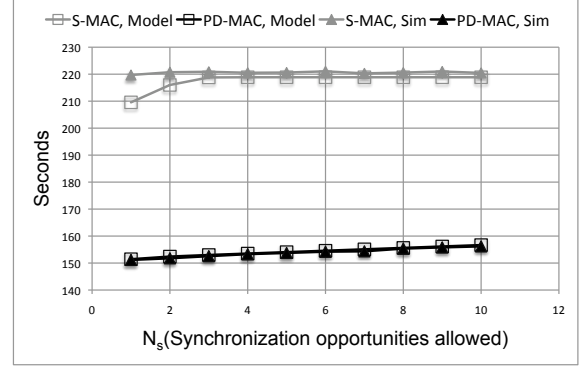


Fig. 6: Expected round duration vs. number of synchronization opportunities.

C. Computation of energy consumption

To model the energy consumption of a node, we split the time duration of Equations (3) and (8) into time durations of various radio modes; idle, transmit, receive, drowsy and ping. Knowing the power consumed in various radio modes (which is a constant voltage times the current drawn in the radio modes), and multiplying with the durations of the modes we arrive at the model for the energy consumed.

1) *Energy consumption model for S-MAC*: In Equation (6), $T_{\text{sync}}(i)$, consists of three parts given by Equations (9)-(11), each with its own level of transmitter and/or receiver power consumption:

$$T_{\text{idle}}^{\text{sync}}(i) = \frac{1}{2} \left(\left\lfloor \frac{i+1}{2} \right\rfloor T_{\text{DD}} + ((i+1) \bmod 2) \mathbb{E}(Y) - (i+1) T_S \right) + \frac{1}{2} \left(\left\lfloor \frac{i}{2} \right\rfloor T_{\text{DD}} + (i \bmod 2) (T_{\text{DD}} - \mathbb{E}(Y)) - (i+1) T_S \right), \quad (9)$$

$$T_{\text{tx-sync}}^{\text{sync}}(i) = \frac{1}{2} \left\lfloor \frac{i+1}{2} \right\rfloor T_S + \frac{1}{2} \left\lfloor \frac{i}{2} \right\rfloor T_S, \quad (10)$$

$$T_{\text{rx-sync}}^{\text{sync}}(i) = T_{\text{tx-sync}}^{\text{sync}}(i) + T_S. \quad (11)$$

$T_{\text{no-sync}}$ can be split in a fashion similar to Equations (9)-(11). Similarly, T_{data}^k consists of two parts given by Equations (12)-(13), each with its own level of transmitter and receiver power consumption:

$$T_{\text{tx-data}}^k = \sum_{l=1}^{|R_k|} P_D^k(l) \left\{ \sum_{r=1}^{N_d} P_{\text{suc}}^r(l) \cdot r \cdot T_D(l) + P_{\text{fail}}^{N_d}(l) \cdot N_d \cdot T_D(l) \right\} = T_{\text{rx-data}}^m, \quad (12)$$

$$T_{\text{tx-ack}}^m = \sum_{l=1}^{|R_k|} P_D^k(l) \left(\sum_{r=1}^{N_d} P_{\text{suc}}^r(l) \cdot r \cdot T_A^1 + P_{\text{fail}}(l) \cdot N_d \cdot T_A^1 \right) = T_{\text{rx-ack}}^k. \quad (13)$$

2) *Energy consumption model for PD-MAC*: Similar to the energy computation for S-MAC, the energy computation for PD-MAC is achieved by splitting the time duration terms in Equation (8) into parts during which the power level remains constant. The duration T_{SYN} denotes the ping duration during which node m is in ping mode while an upstream node $k \in U_m$ is either in drowsy mode (if it has not been synchronized yet) or sleep mode (if it has already been synchronized and expended all its data attempts).

Next we focus on splitting \hat{T}_{DA}^m into durations where the power level remains constant for nodes $k \in U_m$ and m . Let L denote the set of upstream nodes that are newly synchronized and transmit data during a particular data attempt phase captured by \hat{T}_{DA}^m of Equation (8). In case 2: $L = R_m$, while in case 3: $L = Q_m \setminus D_m$. For $k \in L$, \hat{T}_{DA}^m can be split into three parts given by Equations (14)-(16):

$$\hat{T}_{\text{tx-data}}^k = \sum_{l=1}^{|R_k|} P_D^k(l) \cdot T_D(l), \quad (14)$$

$$\hat{T}_{\text{rx-ack}}^k = T_A(|U_m|), \quad (15)$$

$$\hat{T}_{\text{idle}}^k = \hat{T}_{DA}^m - \hat{T}_{\text{tx-data}}^k - \hat{T}_{\text{rx-ack}}^k. \quad (16)$$

An upstream node $k \in U_m$ that has not been synchronized yet spends the entire duration in \hat{T}_{DA}^m in drowsy mode. This happens in case 2 when: $k \in U_m \setminus S_m \setminus R_m$, while in case 3 when $k \in U_m \setminus S_m$. In all other cases, an upstream node k spends the entire duration \hat{T}_{DA}^m in sleep mode, because it has either successfully transmitted its data or expended all its data attempts allowed.

Downstream node m , expends energy, during the period \hat{T}_{DA}^m in idle/transmit/receive modes, until $D_m = U_m$, at which point it goes to sleep mode. If $D_m \neq U_m$, \hat{T}_{DA}^m can be split into three parts given by Equations (17)-(19):

$$\hat{T}_{\text{rx-data}}^m = \sum_{k \in L} \hat{T}_{\text{tx-data}}^k, \quad (17)$$

$$\hat{T}_{\text{tx-ack}}^m = T_A(|U_m|), \quad (18)$$

$$\hat{T}_{\text{idle}}^m = \hat{T}_{DA}^m - \sum_{k \in L} \hat{T}_{\text{tx-data}}^k - T_A(|U_m|). \quad (19)$$

To compute the energy consumption, we used the values for the current drawn from the battery as per the datasheet of CC1110. The results for the total energy consumption of the model as well as the simulations for both S-MAC and PD-MAC are plotted against the number of synchronization attempts in Fig. 7. It should be noted that PD-MAC achieves 65% lower energy consumption compared to S-MAC. This is a result of the proposed PD-MAC that has alternative power modes for radios during wake-up, and also its clever

TABLE I: Current drawn for various power modes

Radio mode	Power Level (dBm)	Current drawn (mA)
Transmit	-30 (Normal)	15
Transmit	10 (Ping)	33.5
Receive	-110 (Normal)	19.8
Receive	-107 (Drowsy)	10.0 (assumed)

exploitation of the converge-cast nature of communication to minimize the number of high energy pings.

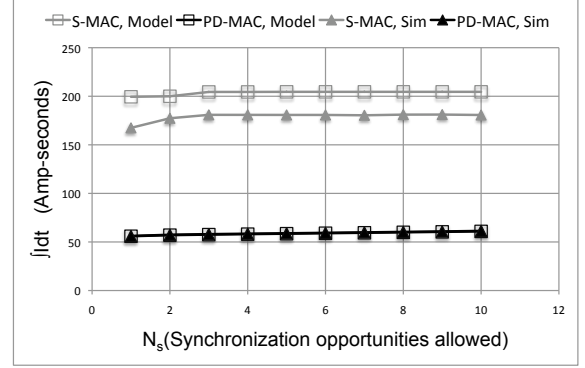


Fig. 7: Expected energy consumption per node.

IV. CONCLUSION

We have presented a performance modeling approach based on probabilistic analysis to compare metrics such as energy consumption, communication delay and throughput of a wireless sensor network employed for a periodic monitoring application. We also compared the performance of our protocol with S-MAC under the settings of our application. While the two protocols can achieve the same packet success rates, the proposed new protocol is able to accomplish this in 25% less time and 65% less energy (as analyzed and simulated for a simple sensor field of 5×5 nodes).

REFERENCES

- [1] H. Sahota, R. Kumar, A. Kamal, and J. Huang, "An energy-efficient wireless sensor network for precision agriculture," in *Proc. IEEE Symposium on Computers and Communications*. Riccione, Italy: IEEE Computer Society, Jun. 2010, pp. 347–350. [Online]. Available: <http://dx.doi.org/10.1109/ISCC.2010.5546508>
- [2] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004. [Online]. Available: <http://dx.doi.org/10.1109/TNET.2004.828953>
- [3] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," in *Proc. IEEE Symposium on Computers and Communications*, Alexandria, Egypt, 2004, pp. 244–251.
- [4] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proc. SenSys '03*. Los Angeles, CA, USA: ACM, 2003, pp. 171–180. [Online]. Available: <http://doi.acm.org/10.1145/958491.958512>