Performance Analysis of Polling based TDMA MAC Protocols with Sleep and Wakeup Cycles

Haiming Yang and Biplab Sikdar
Department of ECSE, RPI, Troy NY 12180 USA

***Abstract*— In sensor networks, MAC protocols based on Time Division Multiple Access (TDMA) with wakeup and sleep periods have attracted considerable interest because of their low power consumption and collision free operation. In this paper, we develop analytic models to evaluate the performance of such protocols. The model presented in this paper characterizes the queueing delays associated with the MAC layer as well as the energy consumed at MAC layer, by modeling the system as a queue with general service time distribution with both polling and vacations. The analysis is validated by comparison with simulation results using the NS-2 simulator. Our results show that polling based TDMA with sleep and wakeup cycles has lower delays and lower energy consumption as compared to SMAC.**

I. INTRODUCTION

The fundamental task of MAC protocols is to schedule transmissions from stations sharing the same channel and prevent collisions, thereby leading to higher throughput and bandwidth utilization. Since wireless sensor networks (WSNs) typically consist of a large number of energy constrained nodes with limited on board battery resources, MAC protocols for WSNs have the additional requirement to be energy efficient.

The energy consumed by the communication circuitry during idle listening [7] represents a major factor to be considered in the energy wastage of MAC protocols. Typical deployment scenarios for WSNs such as monitoring and tracking are associated with intermittent data reporting for the sensor nodes and large periods of inactivity. Most recent proposals for MAC protocols for WSNs exploit this observation by proposing the use of sleep and wakeup cycles to reduce the energy wasted on idle listening. On the other hand, however, the sleep and wakeup cycles in these MAC protocols increases the MAC layer delays. This energy-delay tradeoff is extremely important in the design of such MAC protocols. To address this issue, in this paper we develop an analytic model to evaluate the performance of a class of TDMA based MAC protocols with sleep and wakeup cycles, in terms of both its energy consumption as well as delay characteristics and the associated tradeoffs.

In this paper, we consider the average packet delay and energy consumption of MAC protocols which use TDMA with polling and sleep wakeup cycles and compare its performance with decentralized, SMAC type protocols. The main contribution of the paper is that we develop an analytic model for evaluating the performance of such protocol in terms of its energy consumption and the MAC layer delays. We also show that TDMA with polling and sleep-wakeup cycles can outperform SMAC in terms of both the energy consumption as well as the delays. Our analysis has been verified using extensive simulations.

The rest of the paper is organized as follows. Section II describes the related work on performance modeling of MAC protocols and the details of the TDMA protocol with polling and sleep-wakeup cycles considered in this paper. Section III describes the analytic framework to evaluate the protocol's performance. Validating simulation results and comparison with SMAC are presented in Section IV. Section V concludes our paper.

II. BACKGROUND AND RELATED WORK

A number of MAC protocols have been proposed for wireless sensor networks. The most common of these is based on carrier sense multiple access with collision avoidance (CSMA/CA) [1]. In CSMA/CA, terminals transmit data if the medium is sensed idle, and use a backoff mechanism in case of busy channel or collisions, such as in IEEE 802.11 and SMAC [2]. Additionally, the ready-to-send and clear-to-send (RTS/CTS) handshake mechanism proposed in [6] may be used to avoid hidden terminal problems.

SMAC is one of the most popular MAC protocols for WSNs and is based on the CSMA/CA mechanism. In SMAC each node follows a periodic sleep-wakeup cycle. In the wakeup state, each node first synchronizes with its neighboring nodes and then exchanges any information that it may have. The sleep state is used to reduce the energy consumption.

An analytical model has been developed in [3] to compute the throughput of 802.11 DCF assuming a finite number of terminals with saturated load and ideal channel conditions. In [4], an analytic model for evaluating the queueing delays at nodes using the IEEE 802.11 Point Coordination Function (PCF) MAC has been developed. The performance of SMAC and its variants have been evaluated using simulations in [2], [9], [10].

Due to its benefits of reduced collisions, scalability and bounded latency, TDMA is widely considered in wireless networks. TDMA partitions time into many fixed slots and nodes transmit data in their assigned slots, thereby avoiding collisions. TDMA based protocols are more energy efficient, and the energy consumed is proportional to the length of the transmission cycle while the latency is proportional to the size of the network. Many models for evaluating the performance of traditional TDMA systems have been presented in [11]. In sensor networks, sleep and wakeup cycles may be used in conjunction with traditional TDMA. In such systems, slots are

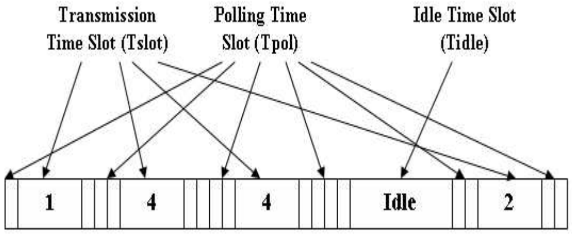

Fig. 1. Protocol operation showing several cycles of transmission

assigned to nodes only after polling them resulting in dynamic assignment of slots on a frame-by-frame basis.

A. Protocol Description

The polling TDMA based MAC protocol with sleep and wakeup cycles considered in this paper assumes that the sensors use a cluster based network architecture since clustering mechanisms have been widely proposed for WSNs to improve their scalability and self-organization [8]. In such mechanisms, sensor nodes in a geographical region select a node amongst them as the cluster head, which is responsible for communicating with other cluster heads and the sink. All other nodes are leaf nodes, and can only communicate with the cluster head in their cluster. The operation of the MAC protocol can be described in terms of cycles. Each cycle begins with the polling of the first node in the cluster. If the node has any packets to transmit, the cluster head immediately assigns a transmission slot for this node. Once the transmission is over, the cluster head continues with the polling of the second node and so on till all the nodes have been polled and the first node is then polled again, marking the start of a new cycle. The length of a cycle is not fixed and may vary from cycle to cycle since only nodes with data to transmit are assigned slots. If a situation arises that all nodes are polled and none of them have any data to to send, in order to save power, the cluster head assigns a sleep period for all nodes.

Figure 1 shows an example of the protocol's operation over several transmission cycles for a cluster with four leaf nodes, numbered 1 through 4. In cycle 1, only nodes 1 and 4 have data to send. Node 1 is polled first and is allocated a transmission slot. Nodes 2 and 3 are then polled and not assigned slots. Node 4 is then polled and assigned a slot. In cycle 2, only node 4 has data to send when the nodes are polled and thus the cluster head assigns a transmission slot to node 4 only. In cycle 3, none of the nodes have any data to transmit and at the end of the polling period, all nodes enter the sleep state for an idle period. At the end of the idle period the nodes are polled again and in cycle 4, only node 2 has data to send. Thus after nodes 1 and 2 are polled, a transmission slot is assigned to node 2. Note that the length of the cycles may vary depending on the number of active nodes.

III. PERFORMANCE MODEL

In this section we present analytical models for evaluating the protocol's MAC layer packet delays and energy consumption.

A. Delay Analysis

We assume that there are M nodes in the cluster, with one cluster head and $M-1$ leaf nodes. The time to transmit one byte on the channel is assumed to be T seconds. The duration of a polling slot is denoted by T_{pol} and during each polling slot, the cluster head transmits k_{pol_dl} bytes to the polled node and the polled node replies using k_{pol_ul} bytes. We use the notation $k_{pol} = k_{pol\_dl} + k_{pol\_ul}$ and thus $k_{pol}T = T_{pol}$. The data generated by each node is assumed to be of k_{slot} bytes, requiring $k_{slot}T = T_{slot}$ seconds to be transmitted. The length of an idle period is denoted by T_{idle}. In each cycle, the length of the polling interval is MT_{pol} seconds which include $M-1$ polling slots for the $M-1$ leaf nodes and one slot for the cluster head to disseminate the synchronization and management information. For each node, we denote its utilization factor by ρ. For an arbitrary arrival in any queue, we denote the number of packets it sees waiting in the cluster by N. Also, at the instant of any arbitrary arrival, the MAC protocol may be in a polling slot, data transmission slot or in an idle period. We use T_{res} to denote the average time left to finish the ongoing polling slot, data transmission slot or idle time, once a packet arrives at the queue. T_{wait} denotes the average waiting time for the packet before it starts its transmission and T_{tran} is the total time it spends in the system.

In this paper, we assume that packets are generated at each node according to a Poisson process with rate λ to facilitate analytical tractability. Also, the Poisson model can capture the randomness and burstiness associated with the traffic generated by some WSN applications. In Section IV-B, we have demonstrated the closeness with which the analytic results using the Poisson assumption matches the results generated from simulations with non-Poisson traffic processes. Specifically, we have compared and shown the close match between the analytical results with simulations using on-off traffic with Pareto distributed on and off times as well as CBR traffic. One of the main aims of this paper is to provide insights into the protocol's performance and understanding the impact the system parameters have on it. Towards this end, the Poisson assumption does not affect the validity of the conclusions.

Our analytic model is based on the M/G/1 queue with both polling and vacations. Models for the classical TDMA system with polling and TDMA with vacation in existing literature [11] is not applicable in our case because we have to consider the vacation periods and polling mode of operation in the queuing model *at the same time*, which makes the analysis quite different.

We now derive the expression for the expected packet delay. With $M-1$ leaf nodes, the overall data arrival rate in the whole cluster is $(M-1)\lambda$. Since it takes T_{slot} seconds to transmit a packet, the service rate is given by $\mu = \frac{1}{T_{slot}}$ and the utilization factor is

$$\rho = \frac{(M-1)\lambda}{\mu} = T_{slot}(M-1)\lambda \qquad (1)$$

Consider an arbitrary packet arrival into the system, regardless of the node it arrives at. The expected delay for this packet consists of three terms: first, the mean residual time T_{res} for the ongoing packet transmission slot, idle period, or polling slot in progress at the instant of the packet's arrival to finish; second, the expected time to transmit the $E[N]$ packets that arrived before this packet; and third, the expected duration of all the polling intervals from the instance of the packet's arrival till its departure, T_Y. Thus, the expected waiting time is given by

$$T_{wait} = T_{res} + \frac{E[N]}{\mu} + T_Y \tag{2}$$

Using Little's law, the expected number of packets in the queue seen by the arrival is given by $E[N] = \lambda(M-1)T_{wait}$ or equivalently, $E[N]/\mu = \rho T_{wait}$. Thus we have,

$$T_{wait} = T_{res} + \rho T_{wait} + T_Y = \frac{T_{res} + T_Y}{1-\rho} \tag{3}$$

We now calculate the residual time T_{res} using mean value analysis. When a packet arrives, the system can be in one of the following three states: idle period, polling period or data transmission slot and the probability that the packet arrives in each of these states is p_{idle}, p_{pol} and p_{data}, respectively. Note that since ρ denotes the fraction of time the system spends in transmitting data, we have $p_{data} = \rho$. p_{idle} and p_{pol} are evaluated later in this section. Also,

$$p_{data} + p_{idle} + p_{pol} = 1 \tag{4}$$

Since data packets arrive according to a Poisson process, the inter-arrival times are exponentially distributed. Also, *given* that an Poisson arrival occurs in an interval or time slot, the instant of arrival of the packet relative to the start of the interval or time slot follows an Uniform distribution [11]. Since the duration of an idle period, a polling slot and a data transmission slot are T_{idle}, T_{pol} and T_{slot} respectively, the expected value of T_{res} is given by

$$T_{res} = p_{idle}\frac{T_{idle}}{2} + p_{pol}\frac{T_{pol}}{2} + p_{data}\frac{T_{slot}}{2} \tag{5}$$

T_Y is the expected duration of all the polling periods which a packet must wait before being transmitted. Since an arbitrary arrival is equally likely to arrive at any of the $M-1$ leaf nodes, the arrival joins the queue of any one of the nodes with probability $1/(M-1)$. Therefore, in steady-state, the expected number of packets which are waiting in the queue of the node is given by Little's law as

$$\frac{E[N]}{M-1} = \lambda T_{wait} \tag{6}$$

and the packet has to wait for a polling cycle for each of these packets. Consider the number of polling slots till the first of these λT_{wait} packets gets served. Irrespective of whether the tagged packet arrives during an idle period, a polling slot or a transmission time slot, an average polling cycle consisting of $M/2$ polling slots needs to pass before the first packet from the queue is served. Note that we have M polling slots in each cycle even though there are $M-1$ leaf nodes since the additional slot is assumed to be used for management purposes by the cluster head. The remaining $\lambda T_{wait} - 1$ packets and the tagged packet itself have to go through polling cycles of M polling slots each before they are transmitted. Combining these observations and noting that the length of each polling slot is T_{pol}, we have,

$$\begin{aligned} T_Y &= (p_{idle} + p_{pol} + p_{data})\frac{MT_{pol}}{2} + \lambda T_{wait} M T_{pol} \\ &= \frac{MT_{pol}}{2} + \lambda T_{wait} M T_{pol} \end{aligned} \tag{7}$$

Before we obtain the expressions for p_{idle} and p_{pol} we first need to evaluate the average cycle length, T_{cycle}. Since the probability mass function for the number of arrivals in the system obeys the Poisson distribution, during an average cycle, we have

$$p(k) = \frac{(\lambda(M-1)T_{cycle})^k e^{-\lambda(M-1)T_{cycle}}}{k!} \tag{8}$$

where $p(k)$ denotes the probability that k packets arrive in an interval of T_{cycle} seconds. The expected number of arrivals in an average cycle is thus $\lambda(M-1)T_{cycle}$. Under steady-state for stable systems, the expected number of arrivals in a cycle equals the number of departures. Thus, each cycle includes an average of $\lambda(M-1)T_{cycle}$ packet transmissions, requiring $\lambda(M-1)T_{cycle}T_{slot}$ seconds. In addition, each cycle includes a polling period of length MT_{pol}. Also, in case the system moves to an idle period, the idle time of T_{idle} gets added to the cycle time. Note that for the system to be stable in the steady state, no more than one packet should arrive at a node during a cycle, on average. Using this observation, we approximate the probability that the system moves into an idle period (of length T_{idle}) by considering the following two cases: (1) there are no arrivals to the system in the previous cycle (which happens with probability $e^{-\lambda(M-1)T_{cycle}}$) and (2) any arrival at a particular node arrives before the node in polled in the cycle (which happens with probability $\frac{\lambda M T_{cycle}}{2}e^{-\lambda(M-1)T_{cycle}}$). Thus the expected length of a cycle is given by

$$\begin{aligned} T_{cycle} &= MT_{pol} + (1 + \frac{\lambda M T_{cycle}}{2})e^{-\lambda(M-1)T_{cycle}}T_{idle} \\ &\quad + \lambda(M-1)T_{cycle}T_{slot} \end{aligned} \tag{9}$$

The fraction of time spent in for polling is thus

$$p_{pol} = \frac{MT_{pol}}{T_{cycle}} \tag{10}$$

and from Eqn. (4) the probability that an arrival occurs during an idle time slot is given by

$$p_{idle} = 1 - p_{data} - p_{pol} = 1 - \rho - \frac{MT_{pol}}{T_{cycle}} \tag{11}$$

Substituting the value for T_Y from Eqn. (7) in Eqn. (3), the expected waiting time is then given by

$$T_{wait} = \frac{T_{res} + \frac{MT_{pol}}{2}}{1 - \lambda(M-1)T_{slot} - \lambda M T_{pol} T_{slot}} \tag{12}$$

The total time spent in the system by a packet is thus

$$T_{trans} = T_{wait} + T_{slot} \tag{13}$$

From Eqns. (12) and (13), we obtain

$$T_{trans} = \frac{T_{res} + \frac{MT_{pol}}{2}}{1 - (M-1)\lambda T_{slot} - \lambda M T_{pol} T_{slot}} + T_{slot} \tag{14}$$

To evaluate T_{trans} and T_{wait} we need the values of p_{pol} and p_{idle} which in turn are dependent on the value of T_{cycle}. Rewriting Eqn. (9), T_{cycle} can be expressed as

$$T_{cycle} = \frac{MT_{pol} + T_{idle}(1 + \frac{\lambda M T_{cycle}}{2})e^{-\lambda(M-1)T_{cycle}}}{1 - \lambda(M-1)T_{slot}} \tag{15}$$

Substituting the solution for T_{cycle} from Eqn. (15) (which can be expressed in the form of a Lambert W function and omitted here for brevity) in the expression of T_{tran} in Eqn. (14), we obtain the following expression for the expected time that a packet spends in the system:

$$\begin{aligned} T_{tran} \; = \; & \frac{\frac{1}{2}T_{idle}\left(1 - \lambda(M-1)T_{slot} - \frac{MT_{pol}}{T_{cycle}}\right)}{1 - \lambda(M-1)T_{slot} - \lambda M T_{pol} T_{slot}} \\ & + \frac{\frac{1}{2}MT_{pol} + \frac{1}{2}\lambda(M-1)T_{slot}^2}{1 - \lambda(M-1)T_{slot} - \lambda M T_{pol} T_{slot}} + T_{slot} \end{aligned} \tag{16}$$

B. Energy Analysis

Using the expressions from the delay analysis developed in the previous section, we now obtain the expression for the rate at which energy is consumed by the leaf nodes. The energy consumption calculations depend on the assumptions about the energy dissipation characteristics of the radio in the sending and receiving modes. While our model can accommodate any radio specifications, for purposes of illustration, we assume that the radio dissipates $E_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $E_{amp} = 100pJ/m^2$ for the transmitter amplifier to achieve an acceptable signal to noise ratio [8]. Assuming r^2 energy loss in the channel, to send a k bits message to a distance d using this radio model, the radio expends:

$$E_{Tx}(k,d) = kE_{elec} + kd^2E_{amp} \tag{17}$$

In order to receive this message, the radio expends:

$$E_{Rx}(k,d) = kE_{elec} \tag{18}$$

We denote the rate of energy consumption in the polling period, data transmission slots and the idle period by E_{pol}, E_{data} and E_{idle} respectively. Now, the fraction of time the system spends in the polling phase is p_{pol}. Similarly, the fraction of time spent by the system on transmitting data packets and in the idle period are given by p_{data} and p_{idle}, respectively. Since the length of a cycle is not fixed and the system enters the idle period if there is a sequence of $M-1$ unsuccessful polls, each leaf node has to stay awake at each polling slot to determine if an idle period is beginning at the end of the polling slot. The average rate of energy consumption by the leaf nodes is thus

$$E_{avg} = p_{pol}E_{pol} + p_{idle}E_{idle} + \frac{1}{M-1}p_{data}E_{data} \tag{19}$$

where the factor of $M-1$ appears in the term for E_{data} since $p_{data}/(M-1)$ reflects the fraction of time that a tagged leaf node spends on transmitting data. The energy consumed by the nodes in the idle period is multiple orders of magnitude lower than the energy consumed in the active mode and thus E_{idle} can be neglected, i.e., $E_{idle} \approx 0$. The average energy consumed by a leaf node is then given by

$$E_{avg} = p_{pol}E_{pol} + \frac{1}{M-1}p_{data}E_{data} \tag{20}$$

Now, since k_{slot} bytes are transmitted by a node in T_{slot} seconds during a data slot, the rate of energy consumption during data transmissions is given by

$$E_{data} = \frac{E_{elec}k_{slot} + E_{amp}k_{slot}d^2}{T_{slot}} \tag{21}$$

Recall that during a given node's polling slot, the node receives k_{pol_dl} bytes sent by the cluster head to poll it and sends k_{pol_ul} bytes to reply to the poll. In addition, the node also listens to the messages (of length $k_{pol} = k_{pol\_dl} + k_{pol\_ul}$) exchanged during the polling of other nodes. Since each cycle has M polling slots, the average rate of energy consumed by a node in the polling period is given by

$$\begin{aligned} E_{pol} &= \frac{k_{pol\_ul}(E_{elec}+E_{amp}d^2)+k_{pol\_dl}E_{elec}+(M-1)k_{pol}E_{elec}}{MT_{pol}} \\ &= \frac{Mk_{pol}E_{elec} + k_{pol\_ul}E_{amp}d^2}{MT_{pol}} \end{aligned} \tag{22}$$

Substituting the above expressions for E_{pol} and E_{data} in Eqn. (20), the average rate of energy consumption of a leaf node is given by

$$\begin{aligned} E_{avg} \; = \; & p_{pol}\frac{Mk_{pol}E_{elec} + k_{pol\_ul}E_{amp}d^2}{MT_{pol}} \\ & + p_{data}\frac{E_{elec}k_{slot} + E_{amp}k_{slot}d^2}{(M-1)T_{slot}} \end{aligned} \tag{23}$$

Substituting the solution for T_{cycle} from Eqn. (15) in the expression above, we obtain the following expression for the rate of energy consumption in terms of the known parameters T_{slot}, M, T_{pol} and T_{idle}:

$$\begin{aligned} E_{avg} \; = \; & \frac{Mk_{pol}E_{elec} + k_{pol\_ul}E_{amp}d^2}{T_{cycle}} \\ & + \lambda(E_{elec}k_{slot} + E_{amp}k_{slot}d^2) \end{aligned} \tag{24}$$

IV. SIMULATION RESULTS

In this section, we validate our analytic results using simulations and compare the performance of polling based TDMA with sleep and wakeup cycles with SMAC. We implemented the TDMA based protocol in the NS-2 network simulator. The length of each simulation run is 2000 seconds and the rate of data transmissions on the wireless channel is 20Kbps.

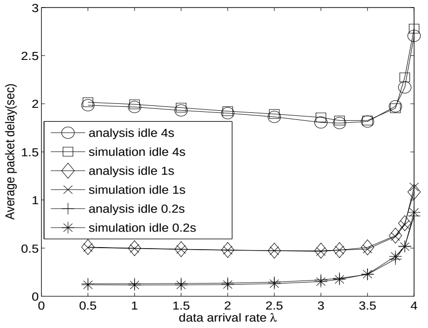

Fig. 2. Average packet delay in a cluster with 9 nodes.

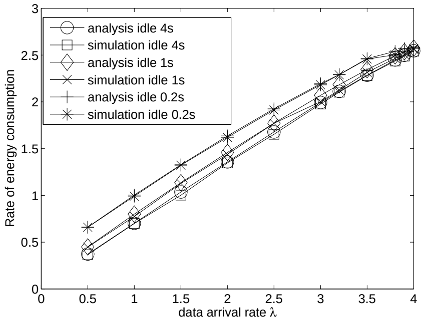

Fig. 3. Average rate of energy consumption in a cluster with 9 nodes

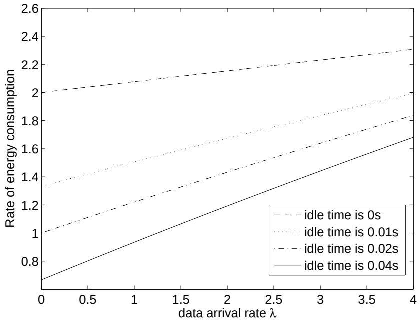

Fig. 4. Blow up of energy consumption rate in the 9 node case.

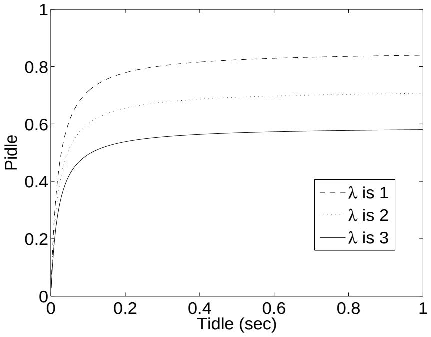

Fig. 5. p_{idle} versus T_{idle}.

Also, $T_{pol} = 0.004$ sec and $T_{slot} = 0.0256$ sec. Results are presented for the cases when there are 9 nodes in the cluster. Results for other cluster sizes are similar.

A. Average Delay and Rate of Energy Consumption

We first validate our analysis for the average packet delay and the rate of energy consumption which are given by Eqns. (16) and (24), respectively. Figures 2 and 3 show the corresponding results for idle periods of 4 sec, 1 sec and 0.2 sec. We note that the simulation and analytic results show a close match in all cases, validating the analytic model.

From the results for the average delays, it can be seen that as expected, the delay increases with the idle time. One key observation from these results is that for larger idle times, the minimum delay is not achieved at low traffic loads but at moderate loads. Also, the minimum value is achieved at an unique arrival rate. The reason behind this is that at low loads, the system is in the idle state most of the time and the length of the idle time dominates the delay. As the load increases, a larger fraction of the packets are transmitted without having to wait for an idle period to end, thereby reducing the delays. At high loads, the queueing delay becomes the dominating factor.

From Figure 3 we note that increasing the idle time reduces the energy consumption rates. We plot the energy consumption rates for a subset of the data arrival rates in Figure 4 to better illustrate the difference among different idle times. Note that while the energy consumption rate decreases with increasing idle time, it cannot continue decreasing unboundedly. The reason is that when the idle time becomes longer, a larger number of packets will accumulate at the queue in each idle period and the subsequent active periods also become longer. As a result, the fraction of time the system spends in the idle period, p_{idle}, saturates and approaches a constant. Thus there are no further savings in the energy consumption. This is shown in Figure 5 where we plot p_{idle} as a function of the idle time T_{idle} for various values of λ and note that p_{idle} saturates as T_{idle} increases.

B. Simulations with Non-Poisson Traffic

The assumptions of a Poisson arrival process made in this paper may not hold in many WSNs. To evaluate the effectiveness of this approximation for non-Poisson arrivals, we now compare our model's results with simulations using CBR traffic and on-off traffic. In the on-off traffic model, packets are generated at a constant rate during the on period and no traffic is generated during the off periods. In our simulations we consider Pareto distributed on and off times in order to produce heavy-tailed on and off times.

Figures 6 and 7 compare the analytic results with those from simulations with Poisson, constant bit rate (CBR) and on-off traffic with Pareto distributed on and off times for a 9 node cluster. Here we show results when the average duration of the on and off times is 100msec, and in the Pareto case, the shape parameter is 1.5 and the position parameter is 0. All other simulation parameters are as before and the length of an idle period is kept at 1 sec. For both delay and energy we note that the results for the non-Poisson traffic match very well with the analytic results for low and high arrival rates.

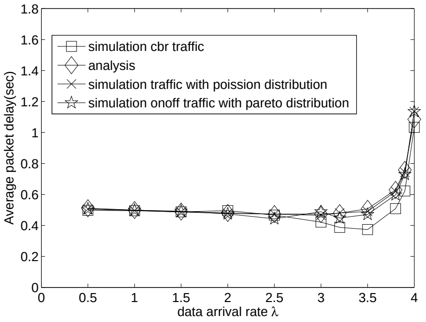

Fig. 6. Delays with non Poisson traffic

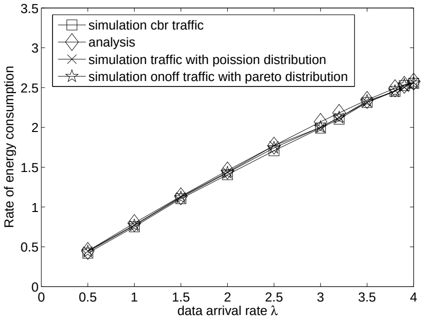

Fig. 7. Energy consumption with non Poisson traffic

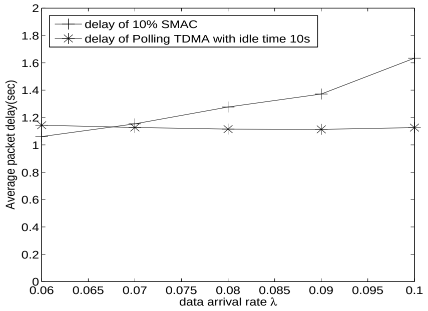

Fig. 8. Packet delays in SMAC and polling TDMA

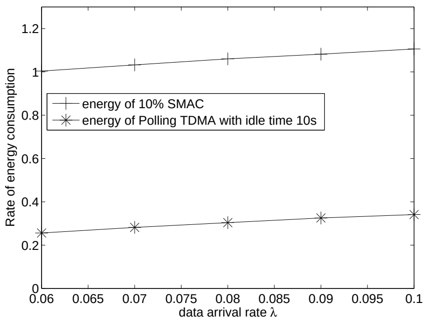

Fig. 9. Energy consumption in SMAC and polling TDMA

Also, the curves show the same trends and at moderate loads, the analytic results are a good approximation.

C. Comparison with SMAC

Since the parameters associated with the two protocols are different, to compare them in similar settings, we first selected the parameters under which both protocols have almost identical performance. We used SMAC with a duty cycle of 10% and a 10 second idle time for polling TDMA so that the delays in both protocols is approximately the same when $\lambda = 0.0675$. Using this point as the baseline, the performance of both protocols is then compared for increasing arrival rates, in terms of the delays and the energy consumption rates, as shown in Figures 8 and 9. We note that polling TDMA outperforms SMAC in terms of both delay and energy consumption. When $\lambda = 0.0675$, the delays of the two protocols are almost at the same as seen in Figure 8 but polling TDMA has around 70% lower energy consumption as compared to SMAC, as seen in Figure 9. The reason is that although both MACs use sleep and wakeup cycle to save energy, SMAC has to spend energy on synchronizing and exchanging packets. Also SMAC uses a fixed wakeup and sleep cycle, and unlike the polling based TDMA which uses adaptive cycle lengths, it does not adapt to the changing traffic conditions, resulting in energy wastage. Also energy is wasted by SMAC through the retransmission which result from its collisions.

V. CONCLUSION

This paper develops an analytic model for evaluating the performance of polling based TDMA MAC protocols with sleep and wakeup cycle. This analytic framework can be used to evaluate the average packet delay and the rate of energy consumption of sensor nodes. Simulation results are used to validate this analytical model and to show that polling TDMA with sleep and wakeup cycle not only achieves lower delays, but also consumes lower energy than contention based SMAC.

REFERENCES

[1] W. Lo and H. Mouftah. “Carrier Sense Multiple Access with Collision Detection for Radio Channels,” *Proc. of IEEE International Communication and Energy Conference,* 1984.
[2] W. Ye, J. Heidemann and D. Estrin, ‘An Energy-efficient MAC Protocol for Wireless Sensor Networks,” *Proc. of IEEE INFOCOM,* 2001.
[3] Giuseppe Bianchi, ‘Performance Analysis of the IEEE 802.11 Distributed Coordination Function” *IEEE Journal on Selected Areas in Communications,* March 2000.
[4] B. Sikdar, ‘Delay Analysis of IEEE 802.11 PCF MAC based Wireless Networks” *IEEE Transactions on Wireless Communications.*
[5] W. Ye, J. Heidemann and D. Estrin, “Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks,” *IEEE/ACM Transaction on Networking,* June 2004.
[6] P. Karn, “MACA - A new Channel Access method for Packet Radio,” *Proc. of ARRL/CRRL Amateur Radio 9th Computer Networking Conference,* 1990.
[7] M. Stemm and R. Katz, “Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices,” *IEICE Transactions on Communications,* Aug. 1997.
[8] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocols for wireless microsensor networks,” *Proc. of Hawaii International Conference on Systems Sciences,* 2000.
[9] J. Ai, J. Kong, and D. Turgut. “An Adaptive Coordinated Medium Access Control for Wireless Sensor Networks,” *Proc. of IEEE ISCC,* 2004.
[10] T. Dam, and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” *Proc. of ACM SenSys,* 2003.
[11] D. Bersekas and R. Gallager, *Data Networks,* Prentice Hall, NJ, 1987.