Random Asynchronous Wakeup Protocol for Sensor Networks

Vamsi Paruchuri, Shivakumar Basavaraju, Arjan Durresi, Rajgopal Kannan and S.S. Iyengar
Louisiana State University
Department of Computer Science
Baton Rouge, LA 70803, USA
{paruchuri,shiva,durresi,rkannan,iyengar}@csc.lsu.edu

Abstract

This paper presents Random Asynchronous Wakeup (RAW), a power saving technique for sensor networks that reduces energy consumption without significantly affecting the latency or connectivity of the network. RAW builds on the observation that when a region of a shared-channel wireless network has a sufficient density of nodes, only a small number of them need be active at any time to forward traffic for active connections.

RAW is a distributed, randomized algorithm where nodes make local decisions on whether to sleep, or to be active. Each node is awake for a randomly chosen fixed interval per time frame. High node density results in existence of several paths between two given nodes whose path length and delay characteristics are similar to the shortest path. Thus, a packet can be forwarded to any of several nodes in order to be delivered to the destination without affecting much the path length and delay experienced by the packet when compared to when forwarded through the shortest path.

Improvement in system lifetime, due to RAW, increases as the ratio of idle-to-sleep energy consumption increases, and as the density of the network increases. Through analytical and experimental evaluations, we show that RAW improves communication latency and system lifetime compared to current schemes.

1. Introduction

Recent advances in wireless communications and microelectro-mechanical systems have motivated the development of extremely small, low-cost sensors that possess sensing, signal processing and wireless communication capabilities. These sensors can be deployed at a cost much lower than traditional wired sensor systems. An ad-hoc wireless network of large numbers of such inexpensive but less reliable and accurate sensors can be used in a wide variety of commercial and military applications such as target tracking, security, environmental monitoring and system control.

In wireless sensor networks, it is critically important to save energy. Battery-power is typically a scarce and expensive resource in wireless devices. This puts significant constraints on the power available for communications, thus limiting both the transmission range and the data rate. Hence, energy efficient communication techniques are essential for increasing the lifetime of such wireless devices.

The design of a good power management protocol for wireless sensor networks needs to consider the following attributes. The first is energy efficiency. Sensor nodes are battery powered, and it is often very difficult to change or recharge batteries for these nodes. Conserving energy and thus prolonging network lifetime for these nodes is a critical issue. Another important attribute is the scalability to the change in network size, node density and topology. Some nodes may die over time; some new nodes may join later; some nodes may move to different locations. A good power management protocol should easily accommodate such network changes. Other important attributes include latency, fairness and bandwidth, which are generally the primary concerns in traditional wireless voice and data networks, but they are secondaryin sensor networks .

Many papers have recently appeared which propose MAC, routing, and topology maintenance schemes that try to save energy based on aggressive power-off strategies. In fact, it has been recognized that the only way a node can save substantial energy is to power off the radio, since transmitting, receiving and listening to an idle channel are functions that require roughly the same amount of power.

In this paper, we present Random Asynchronous Wakeup (RAW), a power management scheme explicitly designed for wireless sensor networks. While reducing energy consumption is the primary goal in our design, our protocol has also achieved good scalability and low latency. To achieve the primary goal of energy efficiency, we reduce idle listening by making the sensors operate at low-duty cycle modes. Low duty cycle increases latency and reduces throughput.

Proceedings of the First International Conference on Broadband Networks (BROADNETS'04)
0-7695-2221-1/04 $ 20.00 IEEE

To reduce latency, RAW uses the concept of Stateless Non-deterministic Geographic Forwarding (SNGF) [6]. Unlike in geographic routing where a packet is forwarded to a node that is closest to the destination, in RAW a packet can be forwarded to any node in the forwarding set as detailed further in the paper. The design reduces the energy consumption due to idle listening and reduces latency because of the presence of multiple forwarding nodes. We present the measurement and evaluation of the trade-offs on energy and latency.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents our proposed protocols. Section 4 describes our simulation model and discusses the simulation results. We present analytical and experimental evaluations of our protocol and compare with asynchronous wakeup protocol [13] in section 5. Section 6 concludes the paper.

2. Related work

There are several solutions addressing the problem of energy wastage due to idle listening. Energy conservation is of paramount importance in sensor networks. The main sources of energy wastage are collisions, idle listening, overhearing and control packet overhead. All MAC [3],[12],[4],[10],[7] protocols contention based (like CSMA) or scheduled protocols (like TDMA) try to avoid collisions. The next major energy wastage source is idle listening, which occurs when the receiver is listening to the channel to receive possible data. As noted earlier, the energy spent during idle listening is comparable to the energy spent during transmitting or receiving. Overhearing occurs when a node receives packets that are destined to other nodes. Overhearing unnecessary packets can be a significant factor in energy wastage when the network is highly loaded or when the node density is high. Lastly sending, receiving and listening for control packets consume energy, which reduces the effective throughput.

One approach to prevent energy wastage due to above sources is to control the node receiver by setting it to sleep mode when no data is expected and to wake up mode when communication is expected (wakeup schemes) [11]. Wakeup schemes can be classified as synchronous and asynchronous. Synchronous wakeup approach is used by the IEEE 802.11 [3] ad hoc power save (PS) mode. This method requires time synchronization of all hosts and considers only single hop communications. Time synchronization in large scale distributed networks such as sensor networks, is generally very costly. Many proposals exist for asynchronous wakeup schemes, wherein each node follows a certain schedule of periodic wakeup and sleep. The final objective of all the schemes is to guarantee the overlap of wakeup times of neighboring nodes within finite time.

An asynchronous wakeup scheme for mobile ad hoc networks by Zheng et al [13], builds on the block design problem in combinatorics. Their scheme consists of two main aspects of neighbor discovery that detect nodes in power saving states, and neighbor schedule book keeping that is used to keep track of neighbor's wakeup schedule to facilitate data communication. In neighbor discovery, each node divides its time axis into fixed length frames of *T* slots. Each slot is, in turn, of length *I*. In this scheme every node chooses the same slot schedule (wakeup schedule function - WSF) to schedule its own active and sleep slots, where WSF is derived from optimal block design. In an active slot, a node can communicate with its neighbors, while in a sleeping slot no communication takes place. A beacon consisting of node *id* and other information for channel contention/ resolution rule signals the start of active slot. Neighbors use beacons to update the neighbor table entries, relative clock and differences in schedules.

The energy savings and wakeup delay can be improved by an additional wakeup or signaling radio. The PAMAS (Power Aware Multi-Access) protocol [8] is an adaptation of the basic mechanisms of IEEE 802.11 to a two-radio architecture. Since the power consumption of the wakeup radio is significantly lower than that of the radio transmission, it can be awake for the entire period, consuming little energy. PAMAS allows a node to sleep to prevent overhearing or to avoid interfering with another's node reception by transmission; however, it ignores the idle listening problem. The main drawback of these schemes is that low power wakeup radio has a lower transmission range than that of radio transmission. This causes limitations where two nodes are within data radio range and not in wakeup radio range. Other than the above problem, two-radio architecture is expensive to implement on sensor nodes that are required to be inexpensive and disposable.

STEM (Sparse Topology and Energy Management) [9] also uses two radios, one is used as a wakeup radio and other is used for data transmission. In STEM, each node periodically turns on its radio receiver for T_{wake} every T duration, where T_{wake}/T is defined as the duty cycle. Low power consumption is achieved by having a high duty cycle ratio instead of low power wakeup radio, thus avoiding some problems discussed above.

S-MAC [12] is a protocol developed to address the energy issue in the sensor networks, building on contention-based protocols like IEEE 802.11. S-MAC follows a simple scheduling scheme that allows neighbors to sleep for long periods and to synchronize wakeups. A complete sleep/wake cycle constitutes a frame. Each frame begins with a listen period for nodes that have data to send. A sleep period follows, during which nodes sleep for a certain period if they have no data to send or receive. Otherwise, they remain awake and exchange data, if they have data to communicate. All nodes independently choose their listen/sleep schedules and share their schedules with neighbors. S-MAC needs synchronization to some extent, but that is not as critical as in TDMA-based protocols. Also, S-MAC uses a fixed sleep interval regardless of traffic load. T-MAC [10] extends S-MAC by adjusting the length of time sensors are awake between sleep intervals based on communication of neighbors. Thus, less energy is wasted due to idle listening when traffic is light.

3 Random Asynchronous Wakeup Protocol

RAW mainly consists of two components - routing based on forwarding sets and random wakeup scheme. The routing methodology in RAW is designed to take advantage of the fact that sensor networks are densely deployed. In conventional routing protocols, the shortest path between two nodes is computed proactively or reactively and a node forwards a packet only to the next node in the shortest path computed. A high node density results in the existence of several paths between two given nodes, whose path lengths are very close to the length of the shortest path. Thus, a packet can be forwarded to any such several pathes in order to be delivered to the destination without affecting the path length and delay when compared to the shortest path. Our random wakeup scheme allows for a node to be active during a randomly chosen fixed interval in each time frame. This removes the necessity of time synchronization and makes the protocol implementation very simple. In this section, we first elaborate our routing methodology based on forwarding sets and then study the random wake up scheme. We then present the complete design of RAW.

3.1 Routing based on forwarding sets

In the geographic routing protocol, a packet is forwarded to a neighboring node that is closest to the destination. However, in a sensor network, in which not all nodes might be active at a given point of time, a packet can be forwarded to the active neighbor that is closest to the destination, or the packet can be queued until the closest neighbor among the rest becomes active, and the packet can then be forwarded to this neighbor.

In this paper, we assume a modification of the geographic routing protocol such that a packet is sent to any of the active neighbors that meet a forwarding criterion (which is discussed later in the section). We define Neighboring Set and Forwarding Candidate Set as follows:

- The Neighbor Set of node *i*: This is the set of nodes that are inside the radio range R of node *i*.

$$NS_i = \{node \mid distance(node, node\ \ i) \leq R\}$$

- The Forwarding Candidate Set of node *i*: For a given destination, this is the set of potential neighboring nodes to which node *i* can forward a packet.

We consider two criteria for defining the Forwarding Candidate Set - one is based on path lengths and the other is based on geographic distance to the destination.

3.1.1 Hop based Forwarding Candidate Set (h-FCS)

Forwarding criterion: For a given source *s* and destination *d*, a neighbor *k* of *s* is a node in FCS if

$$H(k, d) < H(s, d) + \Delta$$

where, $H(i, j)$ is the hop length of the shortest path between nodes *i* and *j*.

When $\Delta = 0$, it implies that a shortest path between *s* and *d* exists through node *k*. When $\Delta > 2$, every neighbor of *s* belongs h-FCS. This is because, for a given neighbor *k*, there always exists a path $s \rightarrow k \rightarrow s \ldots \rightarrow d$ whose length is *H(s, d) + 2*, thus satisfying the forwarding criterion. Also, it should be noted that unless $\Delta = 0$, selecting a forwarding node based on this forwarding criterion does not guarantee that a packet reaches the destination. This is because the path length to the destination from any two neighbors in the path can be same. Figure 1 shows the number of nodes in h-FCS for $\Delta = 0$ and 1 for different densities.

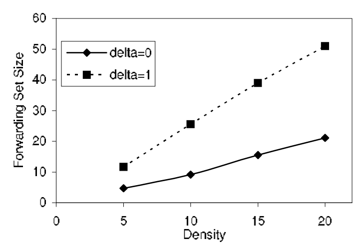

Figure 1. The size of Forwarding Candidate Set for $\Delta = 0$ and $\Delta = 1$

Computing h-FCS requires each node to know the shortest path length to all other nodes in the network. Thus, this criterion of selecting FCS might not be very appealing owing to the computational overhead involved. To overcome this overhead, in the following section, we propose selection of FCS based on the geographic distances between the nodes.

3.1.2 Distance based Forwarding Candidate Set (d-FCS)

Forwarding criterion: For a given source *s* and destination *d*, a neighbor *k* of *s* is a node in FCS if:

$$D(k, d) < D(s, d) - Th$$

where, $D(i, j)$ is the geographic distance between nodes *i*and *j*.

Thus, if a neighbor *k* is closer to the destination by at least *Th* than the node *s* itself, then *k* belongs to the *Forwarding Candidate Set* (see figure 2). The d-FCS selection criterion guarantees that there would be no loops in the path. This is because a node always forwards a packet to a node that is closer to the destination than itself. At the same time, this simple criterion cannot guarantee the delivery of a packet to the destination in presence of holes. At high network densities, it can be safely assumed that holes would not exist. In case holes are present, the criteria for selection has to be extended based on the ideas presented in [5]. In this paper, we assume that no holes are present in the network.

Routing based on forwarding sets increases the path length. The *Th* value limits the maximum path length, as with each transmission a packet traverses at least a distance of *Th* towards the destination. Intuitively, because of increased path lengths, it might seem that Forwarding Set based routing adds additional overhead in terms of energy consumption. However, when combined with the random wake up scheme the total energy consumed by a sensor with RAW is lower.

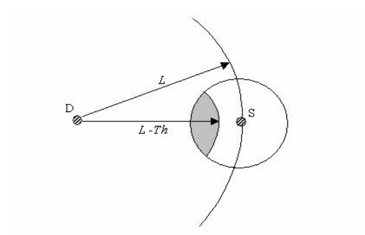

Figure 2. Forwarding Candidate Set (FCS) is set of all nodes lying in the shaded region

3.2 Random Wakeup Scheme

The idea is to have each node wake up once in every slot, be awake for a predetermined time, and then sleep again. To elaborate, consider time slots of fixed interval *T* and the active time of T_a for each sensor node in each time slot ($T_a < T$). Thus, if there are *m* neighbors in the forwarding set of node *S* to which a packet destined to *D* can be transmitted to, then the probability that at least one of those nodes is awake, when *S* is awake is given by:

$$P = 1 - \left(1 - \frac{2T_a}{T}\right)^m$$

Figure 3 shows the probability that at least one node in the forwarding set is active for different T_a values. It should be noted that even for a T_a as low as 15%, at a node density of 10, a node could find an active neighbor to whom it can forward the packet with high probability ($> 82\%$). For higher densities, the probability is even higher. Thus, even if a node is active for a randomly selected duration of Ta, there is a high probability that a packet can be forwarded to the destination. This is used as the basis of design in RAW. The protocol is detailed in the following section.

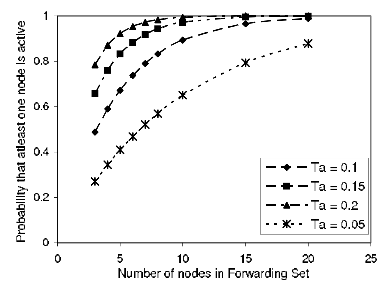

Figure 3. Probability that at least one node in the forwarding set is active for different active times

3.2.1 Neighbor Discovery

The neighbor discovery procedure operates as follows. Whenever a node *i* wakes up, it broadcasts a beacon message piggybacking its own *id*, the start time of its wakeup period and other information subject to channel contention/resolution rule. To implement the protocol, each node keeps a two-hop neighbor list in which each entry has the fields as shown in Table 1.

Table 1. Fields in an entry in a neighbor list maintained by each node

node id	clock	schedule	lifespan	location

A new entry is added whenever a new neighbor is discovered, . Also, among the neighbors of *i*, the node *j* that has been awake for the longest period sends a beacon message to *i* as an acknowledgement, and it also piggybacks its neighbor list. All nodes that receive the acknowledgement beacon update their neighbor lists according to the neighbor list of *j*. This ensures consistency in neighbor lists of all nodes.

3.2.2 Packet forwarding

We use a greedy geographical routing protocol that forwards a packet to an active neighbor that is closest to the destination at each hop. Whenever a node *i* has a packet destined to node *d*, it selects a node *k* from its 1-hop neighbor list, such that *k* is closer to *d* than any other active neighbor of *i* and *k* is closer to the destination by at least *Th*. The threshold *Th* limits the length of a path to a maximum of $D(s,d) * R/Th$. $D(s,d)$ is the distance between the source and the destination, while *R* is the transmission range of the sensors. In section 4.1, we study the effect of *Th* on the performance of the protocol.

4 Performance Evaluation

We have developed a simulator using OMNET++, a discrete event simulation framework [2], to evaluate the performance of our protocol. All simulations were based on a network of dimension $5R \times 5R$, where *R* stands for the transmission range of sensor node. Various node densities were considered. The model parameters and limits on transmission bit rates and energy ratings are set according to Crossbow MICA2 sensor nodes [1]. Power consumption in the model is based on the amount of the current draw that Crossbow MICA2 sensor node's radio transreceiver uses, as shown in Table 2 [1]. We also assume a radio transmission rate of 76.8 kbps.

Table 2. Typical current draw values for simulation purpose

Transmit	Receive	Idle	Sleep
15mA	8mA	7mA	2μA

In our setup, 25 nodes were made to generate traffic to random destinations at the rate 2 packets/sec from every 5 sec. Each data packet had a size of 64bytes including a header of 12 bytes of header information, and hence, the length beacon and other control packets are assumed to be 12 bytes. Nodes were randomly deployed with uniform distribution with densities of 10, 15 and 20 nodes per R2. The energy consumption for switching the radio from idle to sleep modes and vice versa is assumed to be negligible and hence not considered. Also, the location is assumed to be available via GPS or other localization means and thus is not simulated.

We initially study the choice of our protocol parameters: threshold, time frame length and active period. Then, we present the results for average latency per hop, and average delivery ratio at the offered load. The simulation results for each parameter under study are presented in a subsection below. Every simulation is repeated until the 95% confidence intervals for all average results are within $\pm 5\%$.

4.1 Effect of Threshold

The purpose of this study is to evaluate the effect of different threshold values on the performance of RAW. It should be observed that as *Th* increases, the number of neighbors, which are closer to the destination by at least *Th* than the node itself, decreases and, hence, the size of the forwarding candidate set decreases as well. Thus, the probability that a node in the FCS is active decreases and therefore the probability that a packet is buffered, increases. This leads to the increase of latency at higher *Th* values. The effect of *Th* on latency is shown in figure 4(a).

Figure 4(b) shows the effect of *Th* on the delivery ratio. The delivery Ratio is the ratio of packets received at the corresponding destinations to the number of packets generated for the network as a whole. At lower thresholds, though the latency is low, the path length can be very high, thus resulting in many transmissions and collisions.

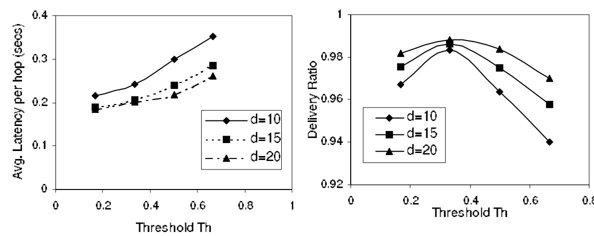

Figure 4. (a) Effect of Threshold Th on Average Latency for various densities (b) Effect of Threshold Th on Delivery Ratio for various densities

At the same time, for a higher *Th*, a packet would be buffered several times, thus increasing the latency and decreasing the capacity of the network. We obtained the best delivery ratios at a *Th* of $R/3$. It should be observed at $Th = R/3$, a maximum delivery ratio of around 98% is achieved. For all further simulations, unless otherwise specified, we use a threshold value of $Th = R/3$.

4.2 Effect of Time Frame length

In this section, we study the criteria involved in setting the time frame length. First, it should be observed that once a node wakes up, it should be active for at least a period that allows the node to transmit the beacon message, receive a reply to the beacon and transmit at least one data packet. The higher the active period of a node, the higher is the frame length, in order to maintain low T_a/T ratio, also known as the duty cycle. This results in higher latency, as a packet will be buffered for longer periods. At the same time, if the active period is short, not many packets can be forwarded and packets might be buffered several times, resulting in higher delay. Figure 5 presents the performance of RAW for different frame lengths. It should be noted that, under the chosen load, the average per hop latency is the least for a frame length of 0.4s. Hence, for all further simulations, we set the frame length to 0.4s unless otherwise specified.

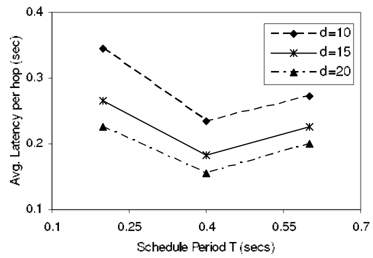

Figure 5. Effect of Schedule Period (T) on the performance of RAW

4.3 Performance of RAW

To validate and evaluate the proposed design, we have conducted a simulation study using different sensor network scenarios. As a baseline, we also evaluate the performance in the absence of power management. The performance metrics of interest are (i) the amount of power consumed, (ii) the packet delivery ratio, and (iii) the latency experienced by the packets.

From the Figure 6(a) it can be observed that the delivery ratio increases with active time. The minimum delivery ratio is observed at a density of 10 nodes/R2 and is always higher than 95%. The figure shows that RAW is scalable with respect to density. In fact, the performance improves with density.

In Figure 6(b) it is shown the effect of T_a on the message latency. The longer a node is active, the lower the latency, and the higher the probability that a node finds a neighbor to which it can forward the packet. The energy consumed by our wakeup scheme for various active durations is presented in Figure 7(a). A density of 10 nodes/R*R is considered. We also present the energy consumption without power management protocol to serve as a base. In Figure 7(b) it is shown the total energy consumed by the network over a simulation of 300s. RAW consumes around 65% less energy than the scheme without power management consumes. The figures also present the trade offs between latency, delivery ratio and energy consumed with the amount of time a node is active. A higher node active time can achieve better latency and delivery ratio, but it will increase the amout of energy consumed. The appropriate choice of node active time depends on the type of application the sensor network is deployed for and the amount of latency and delivery ratio the network can tolerate.

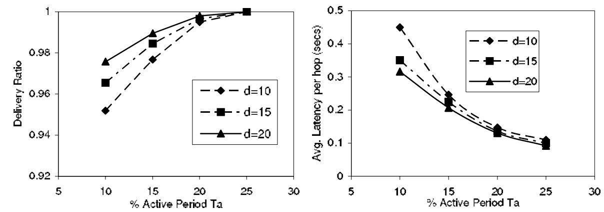

Figure 6. (a) Effect of active time of a node on delivery ratio (b) Effect of active time of a node on average latency

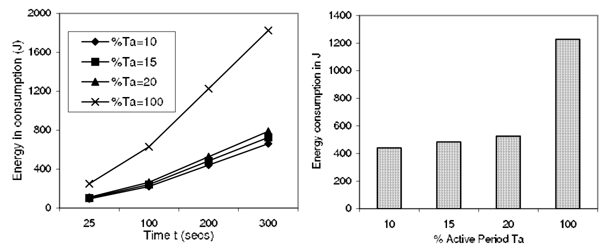

Figure 7. (a) Energy consumed as a function of time (b) Total energy consumed by all nodes in the network

5 RAW and Asynchronous Wakeup

In this section, we present comparisons with the asynchronous wakeup protocol (AWP) [13]. Zheng et al formulate the problem of generating wakeup schedules as a block design problem and derive theoretical bounds under different communication models. Based on the optimal results obtained from the block design problem, they design an asynchronous wakeup protocol, which can detect neighboring nodes in finite time without requiring slot alignment. Hence, we compare our results with AWP that can be easily extended to sensor networks. We consider the performance of the protocols in terms of latency and energy consumed.

We would like to mention that AWP takes a systematic approach to address the protocol design issues of asynchronous wakeup mechanisms for wireless ad hoc networks. While AWP guarantees packet forwarding within one time frame, RAW does not provide such guarantees. Also, it has been shown that AWP performs very well even in mobile networks. However, we do not consider mobile networks in this paper.

5.1 Per-hop latency

In this study, we consider the latency introduced at each node by the wakeup scheduling mechanism. For sake of simplicity, we only consider the delay experienced by a packet due to buffering by the scheduling mechanism. In RAW, each node is active for duration of T_a during each time frame of length *T*. When a node *s* has a packet to forward to *d*, the packet can be forwarded to any node in the forwarding candidate set (FCS). Let *m* be the number of nodes in the FCS. To obtain *m* consider Figure 8.

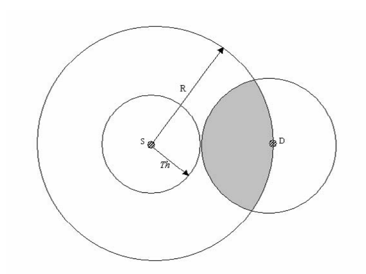

Figure 8. Worst case forwarding area

For a node to be in FCS, it has to be at least *Th* distance closer to the destination. The area where such nodes reside is called the forwarding area. Assuming that the nodes are uniformly distributed inside the system, the larger the size of the forwarding area, the higher is the probability that there will be a node to be chosen. In our protocol such a forwarding area size is not constant; it depends on the distance from the sending node to the destination. The forwarding area is the least when the destination is exactly *R* away from sending node, as shown by the shaded area in Figure 8. The least case forwarding area can be calculated by [6]:

$$A = (R - Th)^2 \cos^{-1} \frac{R - Th}{2R} + R^2 \cos^{-1} \left(1 - \frac{(R - Th)^2}{2R^2}\right) - \frac{1}{2}(R - Th)\sqrt{4R^2 - (R - Th)^2}$$

Now, the probability that some node in FCS is active when *s* wants to forward a packet is given by:

$$P = 1 - \left(1 - \frac{2T_a}{T}\right)^m$$

Thus, with a probability of *P*, a packet is immediately forwarded without any latency being introduced. If *s* cannot find an active node in FCS during its current wakeup slot, it buffers the packet and tries to resend the packet during its next active slot. Thus, the delay encountered by the packet before the node tries to forward it the first time is *T*, which is the average time difference between two consecutive wakeup times. A node keeps buffering the packet until it can forward it. Thus, if the packet is buffered *n* times, the average time a packet is buffered at a node is:

$$Lavg_{RAW} = P \times T \times \sum_{i=1}^{n} (1 - P)^i$$

In AWP, each node divides its time axis into fixed-length frames of *I* slots and each slot is in turn of length *T*. Thus, the total frame duration is $T \times I$. The scheduling algorithm makes sure that between every pair of nodes there is at least common active slot in every frame. The maximum time a packet is buffered is $T \times (I - 1)$, and the minimum time is zero (when both nodes are active the moment packet is received). Thus, the average time a node *i* has to wait before its neighbor *j* is active is $T \times (I-1)/2$. Thus, average delay encountered by a packet at each hop is:

$$Lavg_{AWP} = T \times \frac{I - 1}{2}$$

In Figure 9 are compared the delays encountered by a packet at each hop under RAW and AWP for different parameters. The frame lengths of 0.4 and 0.6 sec, T_a of 10% and 15% are used for RAW, while the (7, 3, 1) and (73, 9, 1) designs are used for AWP. The time frame length in AWP is set to 0.7 seconds. In the (7, 3, 1) design of AWP, a node is awake for approximately 42% of the time while in the (73, 9, 1) design a node is active for 12% of the time. For similar active durations for nodes, the average latency in RAW is significantly lower at intermediate nodes than in AWP.

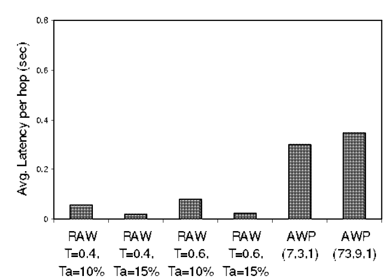

Figure 9. Average latency introduced by RAW and AWP scheduling mechanisms at each node

5.2 Energy consumption

In this section, we compare the performance of RAW and AWP in terms energy consumption. RAW can be easily extended to wireless networks where the bandwidth of nodes is much higher than sensor networks. We have adopted the energy consumption model [13] shown in Table 3 into our simulation model. The raw available bandwidth for each node is set to 2Mbps. We compare our results with the AWP's slot based power management scheme for static networks. In these experiments, we study the energy consumptions of both AWP and RAW schemes under on-off CBR traffic sources. The simulation was performed whith a network of dimension $5R \times 5R$, and with a density of 10 nodes per R^2. In all, 30 connections between random source and destination pairs were used to generate on-off CBR traffic that consists of consecutive periods of *on* and *off* for 10s and 15s respectively. Each simulation run was for a period of 900s with the offered load varying from 5kbps to 40kbps.

Table 3. Power Consumation model for comparison purpose

Transmit	Receive	Idle	Sleep
1400mW	1000mW	830mW	130mW

Figure 10 shows the energy consumed per node per sec under various loads for AWP, RAW and the case without any power management. Two different duty cycles (or % active period T_a) of 10% and 15% were simulated in RAW. Similarly, two different schedules of AWP: (7, 3, 1) design and (73, 9, 1) design were considered. It should be noted that while in the (7, 3, 1) design of AWP a node is active for approximately 42% of the time, in the (73, 9, 1) design, a node is awake for just around 12% of the time, which is similar to active durations in RAW. Understandably, the energy consumption of RAW in both cases is comparable to AWP's (73, 9, 1) design as active times of nodes are similar in these cases.

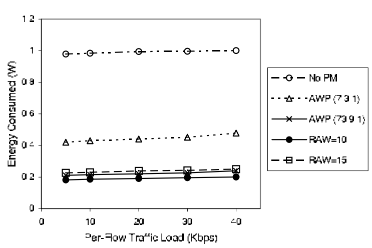

Figure 10. Energy consumed by AWP and RAW

6 Conclusions

This paper presents Random Asynchronous Wakeup, a novel power management protocol for wireless sensor networks. We take a simple randomized approach to address the protocol design issues of asynchronous wakeup mechanisms. The performance of our protocol remains very good even in large networks, and it scales with density. It has very good energy conserving properties while it keeps the latency low. Another interesting property of the protocol is that it has the ability to make trade-offs between energy and latency. Through both analytical and experimental evaluations, we show the improvements of RAW over existing schemes.

References

[1] Crossbow MPR/MIB mote hardware users manual. *www.xbow.com/Support/manuals.htm*.

[2] A discrete event simulation framework OMNET++. *www.omnetpp.org*.

[3] Wireless lan medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard 802.11*, June 1999.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *The Seventh ACM MOBICOM Rome, Italy*, July 2001.

[5] Q. Fang, J. Gao, and L. J. Guibas. An energy efficient MAC protocol for wireless LANs. *IEEE Infocom 2004*, June 2004.

[6] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. *International Conference on Distributed Computing Systems (ICDCS 2003)*, May 2003.

[7] E.-S. Jung and N. H. Vaidya. An energy efficient MAC protocol for wireless LANs. *IEEE Infocom 2002*, June 2002.

[8] C. S. Raghavendra and S. Singh. PAMAS-power aware multi-access protocol with signaling for ad hoc networ. *Computer Communication Reviews*.

[9] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.

[10] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. *ACM Sensys*, November 2002.

[11] X. Yang and N. H. Vaidya. Wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay. *Technical report, CSL, University of Illinois at Urbana-Champaign*, July 2003.

[12] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. *IEEE Infocom 2002*, June 2002.

[13] R. Zheng, J. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 03)*, January 2003.