BASS: an Adaptive Sleeping Scheme for Wireless Sensor Network with Bursty Arrival

Qin Wang
Wireless Network Laboratory
Cornell University
Ithaca, NY 14853, U.S.A
(631)-278-5585
qw29@cornell.edu

Zygmunt J. Haas
Wireless Network Laboratory
Cornell University
Ithaca, NY 14853, U.S.A
(607)-255-3454
hass@ece.cornell.edu

ABSTRACT

Whereas energy efficiency in wireless sensor network is of critical importance, idle listening has been recognized as a main source of wasted power. Many studies have proposed various approaches to scheduling the active and sleeping periods of sensor nodes, as to reduce the power consumption of idle listening. While noticing the fact that there is no universally accepted approach, one which can meet the diversity of different application, in this paper, we propose a scheduling scheme for active and sleeping periods that is based on the packet arrival pattern. More particularly, we propose an arrival model which is targeted at application characterized by bursty arrival. The bursty arrival times are assumed to be distributed exponentially with different rates for the packet arrival intra-burst and inter-burst. Based on this packet arrival model, we introduce a Bursty Arrival Dependent Sleeping Scheduling (BASS) scheme, in which each node dynamically and independently adjusts its wakeup rate. Through analysis and simulation, we evaluate the impact of the proposed scheme on the duty cycle and on the delay of the MAC layer. We show that as the bursty arrival rate decreases, the ON/OFF duty cycle decreases linearly and the MAC-layer delay is minimally affected. Our results suggest that the BASS scheme provides a superior solution for sensor network with bursty arrival. Comparing BASS scheme with S-MAC, results demonstrate 45%-- 70% gains in the BASS case in term of energy efficiency, without degrading performance. BASS scheme also obtains much better performance (e.g. 10 times better in the range of parameters in this paper) with the same amount of power consumption.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols - *Protocol Architecture*

General Terms: Algorithms

Keywords

Wireless sensor networks, Sleeping scheduling, Arrival model

IWCMC'06, July 3-6, 2006, Vancouver, British Columbia, Canada.

1. INTRODUCTION

A wireless sensor network (WSN) consists of a large number of distributed nodes with sensing, data processing, and communication capabilities. Those nodes are self-organized into a multi-hop wireless network and collaborate to accomplish a common task. As sensor nodes are usually battery-powered, and they should be able to operate without attendance for a relatively long period of time, energy efficiency is of critical importance in the design of wireless sensor networks.

Measurements have shown that the energy that a sensor node spends while idly listening amounts to 50%-100% of the energy required for receiving [4]. Furthermore, typically, a sensor node would spend a substantial fraction of the time in the idle state. Therefore, idle listening has been recognized as one of major sources of energy waste in sensor networks and sleep scheduling has been widely studied. The mainstream of research on sleep scheduling can be divided into two approaches. One approach, the "periodical packet-arrival based approach" (e.g., [14] and [12]), assumes periodical packet arrival, thus proposing a periodic active/sleep (i.e., ON/OFF) schedule. The second approach is "coverage-based approach" (e.g., [9], [18], and [15]), which assumes large density of sensor nodes, thus maintaining the connectivity of the network by a subset of nodes which are ON all the time, while letting the other nodes sleep. There are also various strategies for adaptation of the sleeping schedule, that is ending the ON period according to different criteria, such as the overheard messages [14] [12] [17], the network topology [16] [1], the residual energy of the nodes [5], the most recently updated neighbor sleeping schedule [8], the database of neighbor nodes' sleeping schedule [6], the number of packets queued in the MAC layer [2], and the waiting time of packets and the length of waiting queue in the previous node [13].

In spite of the many proposed strategies for adaptation of the sleeping schedule, almost all of these approaches assume that the packet arrival follows a constant-rate Poisson distributed arrival model or the periodic arrival model. Observing the weakness of the two kinds of arrival models, the quasi-periodic arrival model has been recently introduced in studying the sleep scheduling [10]. In addition, a general Pareto arrival process has been introduced in studying traffic anomaly detection [7]. Markov Chain and M/G/1 queue with server vacation model have been used to analyze the performance of sleep scheduling schemes [11] [3].

In this work, we consider practical scenarios of wireless sensor networks, where the packet arrivals are bursty arrivals, and take the sensor networks used for monitoring of bridges, buildings, and equipments in a factory as examples. In such cases, sensors transmit their reading as data packet regularly or as a response to an interrogation and the arrival process shows the characteristic of bursty arrivals. Most of the previously-proposed schemes would not perform well in this scenario because of their assumption of a constant-rate Poisson distributed arrivals or periodic arrival. For instance, for the "periodical packet arrival based approach," during the times inter-burst, nodes would be placed in the active state, wasting energy. On the other hand, during the time the system is intra-burst, excessive delay will occur because nodes would be placed in the sleeping mode. For the "coverage-based approach," during the times inter-burst, the active nodes would waste energy a lot, especially while bursty arrival rate is restively small

In this paper, we propose an approach that is driven by the packet-arrival model. The rest of this paper is organized as follows. Section II describes the bursty arrival model; Section III introduces the BASS algorithm; Section IV evaluates BASS and compare it with a representative periodical packet arrival based approach (S-MAC) through analysis and simulation. Finally, Section V concludes the paper.

2. BURSTY ARRIVAL MODEL

Assume that the sensor nodes are uniformly deployed over a sensing field and at least one sink is placed in the field. For such a monitoring application of a sensor network, there are three possible kinds of data communication patterns:

- ✧ *Scheduled event*. Sensors follow a schedule, measure and then send data packets to a sink regularly at a low rate.
- ✧ *Interrogation event*. One of the sinks interrogates some or all the sensors to collect specific data.
- ✧ *Trigged event*. Sensors, trigged by an event, transmit the sensed data to a sink occasionally.

For the *scheduled event*, sensors send their measured data periodically at a low rate. But because of the difference of clock in each node, the arrival pattern should be modeled as a Poisson packet arrival with a low average arrival rate (denoted as λ_b). For the *interrogation event,* the action of measuring and sending in each node is initiated by a sink, instead of following a preset schedule. So the *interrogation event* can be modeled as a Poisson packet arrival with a high average arrival rate (denoted as λ_a) in a relative short interval, i.e. a bursty arrival. A *trigged event* may cause a bursty arrival or may cause a sparse packet arrival, depending on which sensors are triggered. Thus, in general, Poisson packet arrival with a high average arrival rate λ_a is used to model the bursty packet arrival caused by an *interrogation event* or an *triggered event*, and Poisson distribution with low average arrival rate λ_b is used to model the sparse packet arrival caused by an *scheduled event* or an *triggered event*. Fig.1 shows the arrival model of bursty arrivals in wireless sensor networks.

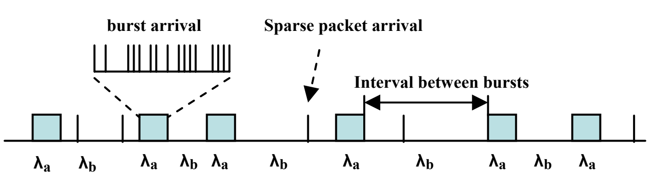


Figure 1. Bursty arrival model

Where, for each node, packet arrival is a Poisson arrival process with $\lambda(t)$. Because $\lambda_a >> \lambda_b$ (typically $\lambda_a/\lambda_b > 100$), the Poisson arrival with rate λ_a dominates the intra-burst period. Thus,

$$\lambda(t) = \begin{cases} \lambda_a & if \quad in \quad busrt \\ \lambda_b & else \end{cases} \qquad (1)$$

Furthermore, because the *interrogation events* are independent, as are the *triggered events,* we assume that the interval between two bursty arrivals is exponentially distribution with parameter λ_c. In addition, the interval between bursts is usually much longer than the interval intra-burst in the monitoring application of wireless sensor networks. Thus, λ_c should be relatively small, and then the length of intra-burst interval can be thought as a relatively small constant.

3. BASS ALGORITHM

3.1 Assumption

We define the timing relationship between sender and receivers as follows. To avoid the difficulty in implementation, the BASS scheme does not require clock synchronization between nodes. Fig.2 shows the asynchronous timing relationship between a transmitter (TX) and different receivers (RXj), with different waking-time offset relative to the TX's waking time. That means TX may wakeup after RX2, at same time as RX1, or before RX3.

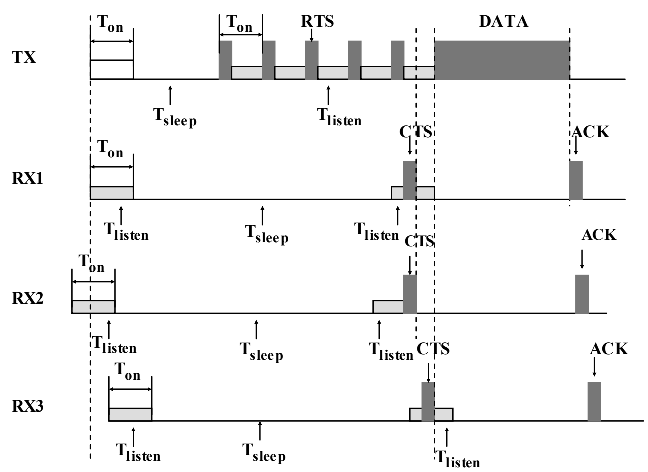


Figure 2. Timing relationship of transmitter vs. different receivers

Here T_{listen} denotes the period of idle listening; T_{sleep} denotes the period of sleeping, T_{on} denotes the period of being active (i.e., ON). For the nodes which hold packets to be transmitted, the T_{on} period includes the time to send the RTS with the total number of data packets, and T_{listen}. For nodes who wait to receive packets, the T_{on} period includes only T_{listen} or T_{listen} and CTS. After the data is received, the receiver sends ACK. There are some cases in which one can guarantee that such an asynchronous timing relationship will work well. For example, if T_{on} period in the receiver side equals to the period of two RTSs plus T_{listen} between the two RTS, it is guaranteed to receive an RTS no matter what is the timing relationship between the transmitter and the receiver. In order to simplify the discussion, we assume that the duration of T_{on} is equal for both, the transmitter and the receiver. We refer to this time as a T_{on} period, denoted as T_{on_period}. Therefore, we can assume that a receiver can deduct that there are waiting packets to be forwarded by listening for RTS, and a transmitter can learn about any active receivers by listening for CTS.

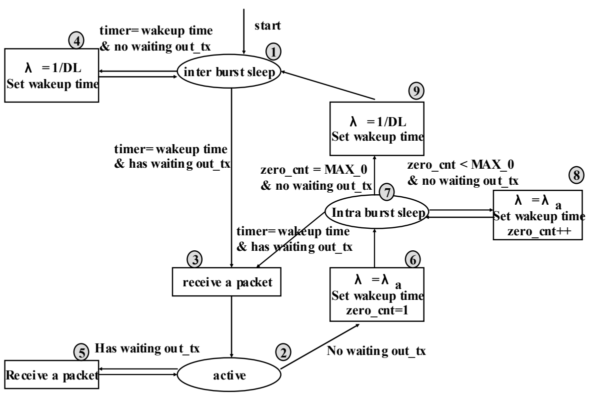
In addition, we define duty cycle as the ratio of the time for idle listening in the duration of the observation to the length of the observed duration, in unit of T_{on_period}.

3.2 Algorithm and Implementation

The fundamental idea of BASS is to schedule the wakeup rate dynamically, according to the characteristics of the arrival rates, i.e. the average intra-burst packet arrival rate is significantly larger than the average inter-burst packet arrival rate.

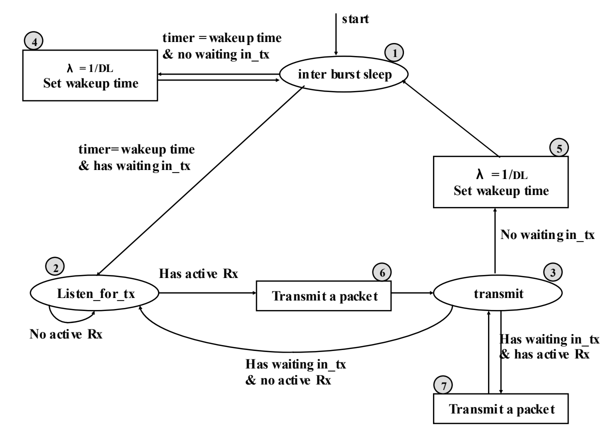
There are two problems with this approach. First, it is difficult to instantaneously detect significant changes in the average arrival rate. Second, if we set the wakeup rate according to the average packet arrival rate, the delay constraint inter-burst is usually difficult to meet, because a small value of λ_b would indicate a long sleep period, possibly violating the delay constraint.

We use the Generalized State Machine (GSM) [1] to describe the algorithm for receiving and for transmitting (Fig.3).



Note: "has waiting out_tx" means having forwarding request from previous nodes.

Figure 3(a). GSM from the view of receiver



Note: "has waiting in_tx" means that the node holds packets to be transmitted.

Figure 3(b). GSM from transmitter view

[1] In the GSM, an ellipse corresponds to a state, a rectangle corresponds to an operation, and an assertion above directed line is a condition for a transition between the states, or from a state to an operation. Absence of an assertion above the directed line from an operation indicates that the transition occurs unconditionally, which means the state pointed by the directed line will be reached just on the time when the operation is finished without any condition. Circled numbers are used for illustration.

Where, the "inter burst sleep" state in both Fig3(a) and Fig.3(b) corresponds to the same state. The two parts of the algorithm run on each node concurrently.

We explain how the above problems are addressed as follows.

(i) Wakeup rate λ is adaptive with the packet arrival rate

(a) In the "inter burst sleep" state, when a node wakes up and receives an RTS which indicates that there is a packet waiting to be received, the node receives the packet and transits into the "active" state

(b) In the "active" state, when the node wakes up and cannot detect an RTS, indicating that there are "no waiting packets," the node sets the next wakeup rate using the average intra-burst packet arrival rate λ_a, and transits into the "intra burst sleep" state.

(c) In the "intra burst sleep" state, a node counts the number of times that it wakes up and finds "no waiting packets", denoted by *zero_cnt*. If the number is less than an upper bound (denoted by *MAX_0*), then the node sets the next wakeup rate using λ_a, and transits into the "intra burst sleep" state again; otherwise, the node sets the wakeup rate with the reciprocal of the maximum delay (denoted by *1/DL*), and transits into the "inter burst sleep" state. We will formally define the "maximum delay" later; it is always much smaller than $1/\lambda_b$.

(d) If already in the "inter burst sleep" state and while waking up, finding that there are "no waiting packets," the node sets the wakeup rate with (*1/DL*), and transits into the "inter burst sleep" state again.

The adaptation scheme for the wakeup rate λ is based on the assumption that λ_b is much less than λ_a. Concretely speaking,

✧ The basic idea behind behavior *(a)* is that any time when the first waiting packet is met, the node assumes that it is the beginning of a burst.

✧ The basic idea behind *(b)* and *(c)* is that when the condition of "no waiting packet" occurs consecutively only a small number of times (less than the upper bound of *MAX_0*), it is likely that the node is in the intra-burst state. Note that *MAX_0* is related to λ_a, and based on our simulation of the optimized duty cycle, for $\lambda_a \geq 0.3$, *MAX_0*=1 and$0.3 > \lambda_a \geq 0.05$, *MAX_0*=2.

✧ The basic idea behind *(d)* is that when the condition of "no waiting packet" is consecutively encountered more than *MAX_0* times, the node is likely in the inter-burst state.

(ii)Listening policy

(e) *intermittent listening* for receiving packets (see "1"=> "4" => "1" in Fig.3 (a)) means that it takes only one T_{on_period} to listen for RTS in order to find if there is a node wishing to send a packet to the node in question;

(f) *continuous listening* to transmit packets (see "2" => "2" in Fig.3 (b)) means that when a node holds a packet to be forwarded, it keeps transmitting RTS and listening for CTS in order to find the successive node wakeup, which guarantees TX and RX's T_{on} period overlap.

The reasons for the listening policy are as follows.

✧ According to the λ-adaptation policy, the listening for transmission occurs mostly in the interval between bursts, and does not happen frequently because of the small value of λ_b. Moreover, the largest number of T_{on_period} that a node needs to wait before transmission of a packet is *DL*, and is not related to the value of λ_b. Thus, it is deduced that *continuous listening* to transmit does not incur substantial energy cost.

- ✧ Because the frequency of holding packets is determined by λ_b, the frequency of finding forwarding request is determined by *1/DL*, and λ_b is always significantly less than *1/DL*, *continuous listening* to transmit is more energy efficient than *continuous listening* to receive. Moreover, since "*continuous listening* to transmit" is used, it is unnecessary that the receiver listens for more than one T_{on_period}.

(iii)Exhaustive receiving/transmitting:

See "2" => "5" =>"2" in Fig.3(a) and "3" => "7" => "3" in Fig.3(b). With *exhaustive receiving* and *exhaustive transmitting* scheme, the process of *continuous listening* to transmit is executed only one time, in order to send all packets in buffer. It cost less energy comparing with such kind of scheme as the packets in buffer are transmitted in several times.

4. EVALUATION

In the following, the analysis and simulation focus on the relationship between the parameters of bursty arrival model and duty cycle, and the relationship between the duty cycle and one hop (i.e., MAC-layer) delay under different parameter setting of arrival model.

4.1 Analysis

Because BASS is an asynchronous sleeping schedule, the number of T_{on_period} in idle time should include both the number of T_{on_period} in idle time waiting for receiving and the number of T_{on_period} in idle time waiting for transmitting. We use N_{l_r} and N_{l_t} denote them respectively. In addition, *DL* is used to denote maximum delay of one hop (MAC layer delay), which equals to the number of T_{on_period} for a packet to wait for transmission in the buffer of the transmission node. *T* is used to denote the observation period in unit of T_{on_period}. *K* denotes the number of T_{on_period} that the previous node finds a packet to be transmitted during the *T* period. N_{dc} is used to denote duty cycle, which is the ratio of the number of T_{on_period} used for idle listening to *T*.

Because of the features of the arrival model described in section II, we assume λ_c is quite small, smaller than 0.01, which means the intervals between bursty arrivals dominate the observation period. Then, instead of measuring the times used for idle listening in whole observation period, we just need to measure the time used for idle listening in the intervals between bursts. Thereby, N_{l_r} equals *T* divided by *DL*, because the wakeup rate in the intervals between burst is set as *1/DL*. Also, the ratio of *K* to *T* is the average packet arrival rate during the inter burst period, i.e. λ_b, and the average number of T_{on_period} while waiting for an active receiver is *DL/2,* because its minimum number of T_{on_period} is zero and its maximum number of T_{on_period} is *DL*. Thus,

$$N_{dc} = \frac{N_{l\_r} + N_{l\_t}}{T} = \frac{(T/DL) + K(DL/2)}{T} = \frac{1}{DL} + \lambda_b \times \frac{DL}{2} \qquad (2)$$

The value of *DL* that minimizes N_{dc}, denoted as DL_0, can be easily obtained from (2) as:

$$DL_0 = \sqrt{2/\lambda_b} \qquad (3)$$

The DL_0 calculated with (3) can be used as an initial value in the simulation later, so as to make simulation converge faster. Table 1 shows the values of DL_0 obtained by equation (3) and those obtained by simulation. Where, assume that λ_a=0.9, λ_c=0.001.

Table 1. DL_0 by calculation vs. by simulation

λ_b	DL_0 by calculation	N_{dc}	DL_0 by simulation	N_{dc}
0.1	4.47	0.2737	10	0.11
0.05	6.32	0.1832	10-15	0.14
0.01	14.1	0.076	10-15	0.121
0.005	20	0.0525	15-20	0.09
0.001	44.7	0.0232	30-35	0.054

We offer the following comments on the above results:

- ✧ When λ_b is either 0.1 or 0.05, the behavior is significantly affected by the behavior during intra-burst times, which are ignored in equation (2). It results in a relatively large difference between the calculated and the simulated values.
- ✧ The value of DL_0 obtained by simulation is usually smaller than the value obtained by calculation. This is because, in simulation, DL_0 is also affected by the behavior of intra-burst intervals, which is simplified in the calculation of DL_0. For instance, DL_0=44.7 may be the best from the view of the inter-burst intervals, but it is too long for the intra-burst interval.

4.2 Simulation

Through simulation, we evaluate the relationship among the duty cycle, the maximum one hop delay, the average one hop delay, and the parameters λ_b or λ_c and compare the performance of BASS with a typical sleeping schedule scheme (S-MAC).

4.2.1 Simulation setting

For setting simulation, it is helpful to clarify the function of the three components such as routing, sleep scheduling, and medium access control (MAC) in the protocol stack. The routing component chooses the forwarding node. The sleep scheduling component implements the ON/OFF duty cycling. The MAC component deals with contention among the multiple nodes that access the medium. Thus, the component of sleep scheduling focuses on scheduling active and sleeping period according to the sequence of packet arrivals, regardless of which node the packets is to be transmitted to or received from. To concentrate on the performance of the sleeping scheduling algorithm, we assumed that CDMA is used in MAC layer to avoid collision, and so that there is no interference between any two connections. In other words, we concentrate on the one-hop network. Moreover, as we mentioned above, from the view of receiver node, the packet arrival pattern is characterized by bursty arrival, thus the virtual queue of packets from senders can be generated with the bursty arrival model described in section II.

Our simulation is divided into two subroutines, one generates the virtual queue of packets according to the bursty arrival model and another executes the BASS algorithm.

A virtual queue of packets, denoted by VQP, is generated following the bursty arrival model described in section II with parameter λ_a, λ_b, and λ_c. The VQP is a sequence over T_{on_period}, i.e. for each T_{on_period}, one of two values is assigned, "no packet" or "packet arrival". The BASS algorithm is driven by the VQP, such that in every T_{on_period}, the following steps are executed:

Step 1. (Transmission) Execute the transmission algorithm as defined in Fig.3(b). Read a value from VQP, buffer the packet if the T_{on_period} is of "packet arrival" value, execute the operations and move from one state to another according to the status of data buffer, the current state of the system, and the result of Step 2 in the previous T_{on_period}.

Step 2. (Receive) Execute the reception algorithm as defined in Fig.3(a). Execute the operations and move from one state to another according to the result of Step 1 and the current state of the system.

During the above execution, we collect performance data, such as the number of T_{on_period} used to wait for transmitting, the number of T_{on_period} used to wait for receiving, the delay distribution, etc.

4.2.2 Results

Fig.4 and Fig.5 shows the performance of BASS in term of duty cycle. Fig.4 shows that for given small enough λ_c, the duty cycle decreases linearly with the decrease in λ_b. Fig.5 shows that for given λ_b, the duty cycle decreases linearly with the decrease in λ_c. The Fig. 5 also shows that a better value of duty cycle can be obtained by setting *DL* according to the value of λ_c

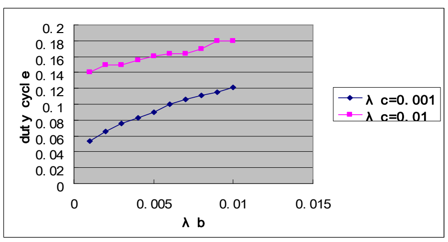


Figure 4. Duty cycle vs. λ_b, Assume λ_a=0.9, λ_c = 0.001, 0.01.

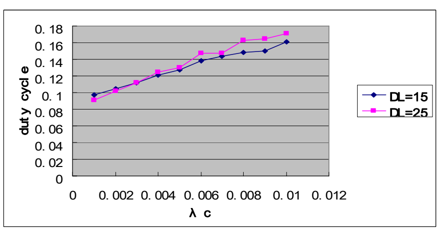


Figure 5. Duty cycle vs. λ_c. Assume λ_a=0.9, λ_b=0.005, *DL*= 15, 25.

In the following, we compare the BASS scheme with the S-MAC scheme. S-MAC is a MAC protocol designed for wireless sensor networks, in which a synchronized sleeping scheduling is proposed. The scheme is based on a periodic schedule of idle listening and sleeping periods. For comparison purpose, we denote duty cycle of BASS as N_{dc_b}, and the duty cycle of S-MAC as N_{dc_s}. N_{l_r} and N_{l_t} represents the number of T_{on_period} in idle time waiting for receiving and for transmitting in BASS, respectively. N_{id} represents the number of T_{on_period} in idle listening in S-MAC. *T* is the duration, in unit of T_{on_period}, of the observation. Then, to compare performance, we define:

$$N_{dc\_b} = \frac{N_{l\_r} + N_{l\_t}}{T} \tag{5}$$

$$N_{dc\_s} = \frac{N_{id}}{T} \tag{6}$$

Firstly VQP is generated with λ_a=0.9, λ_b=0.005, and λ_c=0.005. Then, by executing BASS for different one hop maximum delay (*DL*), the corresponding values of the one hop average delay and the duty cycle are obtained. With the same VQP as input, by executing S-MAC with various ratios of number of T_{on_period} in the active period to that in a time frame of the schedule, the values of one hop maximum delay, average delay and duty cycle are obtained. See Fig.6(a) and Fig.6(b). We note that:

- ✧ For a particular value of one hop maximum delay in BASS, there is an optimized value of the duty cycle. For example, for one hop maximum delay constraint of 15, the optimized duty cycle equals to 0.127.The duty cycle will not benefit from looser one hop maximum delay constraint. For one hop average delay, the situation is similar, i.e. when the duty cycle reaches its optimized value of 0.127, the one hop average delay is about 6.18.
- ✧ Under the same one hop delay constraint as above, the duty cycle in BASS is always smaller than the duty cycle in S-MAC. For example, when the one hop maximum delay constraint is 8, the duty cycle of BASS is 0.15 and the duty cycle of S-MAC is 0.46. In this example, BASS has 63% gains in term of energy efficiency.
- ✧ If the duty cycle is fixed, S-MAC results in much longer one hop maximum delay and one hop average delay than BASS does. For example, when the duty cycle is about 0.25, the one hop maximum delay and one hop average delay of S-MAC is longer than 30 and 9.9, respectively, while the one hop maximum delay and one hop average delay of BASS is about 3 and 1.14, respectively, i.e. BASS has 10 times better performance for this specific example, as compared with the S-MAC scheme for the same amount of power consumption.

The fundamental reason for the above result is that in BASS, each node can adjust the sleeping schedule according to the packet arrival pattern independently and flexibly, while in S-MAC, each node must follow a given and pre-determined sleeping schedule, with energy wastage for idle listening between bursts, and with extra delay when forwarding packets intra-burst.

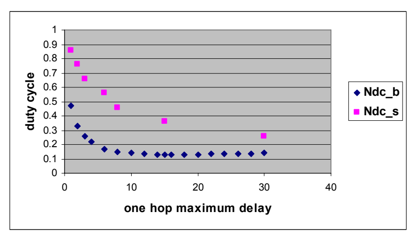


Figure 6(a). Duty cycle vs. one hop maximum delay

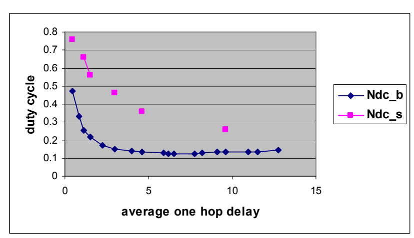


Figure 6(b). Duty cycle vs. average one hop delay

In Fig. 7, we present results of simulation where we examine the range of the parameters of the bursty arrival model, for which BASS outperforms S-MAC and vice-versa. Fig.7(a) and Fig.7(b) show that if other conditions are set as above, when the λ_b and the λ_c is less than about 0.1, the duty cycle of BASS is less than the duty cycle of S-MAC. Thus, indeed, BASS is suitable for the wireless sensor networks characterized by bursty arrival.

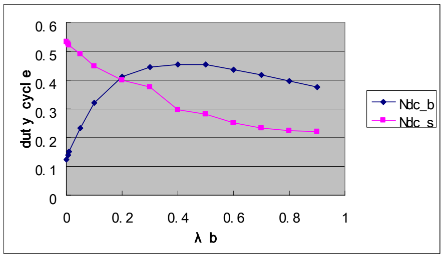


Figure 7(a). Duty cycle vs. λ_b by BASS and S-MAC respectively. Assume λ_a=0.9, λ_c=0.005, and maximum delay is 10 T_{on_period}.

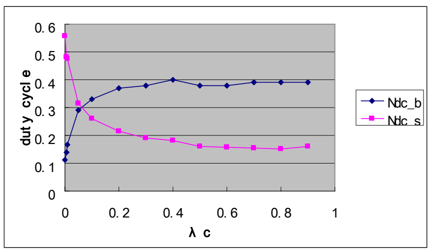


Figure 7(b). Duty cycle vs. λ_c by BASS and S-MAC respectively. Assume λ_a=0.9, λ_b=0.005, and maximum delay is 10 T_{on_period}.

5. CONCLUSION

In this paper, we propose an arrival model for the wireless sensor networks characterized by bursty arrivals. Based on the arrival model, a novel sleep scheduling scheme (BASS) is proposed, in which each node adjusts its wakeup rate to optimize its sleeping schedule independently and dynamically.

The analysis and simulation show that the duty cycle decreases linearly when the average packet arrival rate inter burst decrease; and the duty cycle decreases linearly when bursty arrival rate decreases, and results in only a small change in the one hop delay. This feature of our scheme is useful for a wide range of applications characterized by bursty arrivals.

Comparison of BASS with S-MAC shows 45%-70% gains of BASS in term of energy efficiency, without increasing one hop delay, or, on the other hand, BASS obtains much better performance (10 times in the above example) with the same amount of power consumption. More than it, when the λ_b or λ_c decreases, the energy gain and performance gain will be greater. The comparison also shows BASS is much more suitable for applications of wireless sensor networks characterized by bursty arrival. In addition, the scheme is easy to implement, because the nodes do not need to be synchronized and there is no synchronized sleeping schedule.

In our future research, in order to improve BASS, we plan on addressing the following issues: (1) Estimation of unknown λ_a. (2) More efficient algorithms to determine when a burst starts and when a burst ends? (3) Take advantage of some system features of sensor nodes, e.g., the ratio of the power consumed by Tx/Rx module to power consumed by the sensing and processing module.

6. REFERENCES

[1] Gang Lu, Bhaskar Krishnamachari, Gauligi S. Raghavendra; An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, IPDPS2004

[2] Jing Ai, Jingfei Kong, Damla Turgut. An adaptive coordinated medium access control for wireless network, ISCC2004

[3] K,M.lau and Onching Yue; Modeling synchronous wakeup patterns in wireless sensor network-server vacation model analysis, PMCIA2005

[4] M. stemm, R. H. Katz, “Measuring and reducing energy consumption of network interfaces in hand-held devices”, IEICE Trans. Communication. pp1125-1131, Aug. 1997

[5] Marc R. Pearlman, Jing Deng, Ben Liang, and Zygmunt J. Haas, Elective participation in Ad Hoc network based on energy consumption, GLOBECOM02

[6] Mostafa Ghannad Rezaie, Vahid Shah Mansouri, Mehdi Mani; Critical area attention in traffic aware dynamic node scheduling for low power sensor networks, WCNC2005

[7] Onat, I.; Miri, A, A Real-Time Node-Based Traffic Anomaly Detection Algorithm for Wireless Sensor Networks**,** ICW05

[8] Qing Cao, Tarek Abdelzaher, Tian He, John Stankovic, towards optimal sleep scheduling in sensor networks for rare-event detection, IPSN2005

[9] Rong Zhang, Jennifer C.Hou and Lui Sha, Asynchronous wakeup for Ad Hoc Networks, MobiHoc’03, 2003

[10] Sharat C. Visweswara, Apurva A. Goel, Rudra Dutta, An adaptive ad-hoc self-organizing method for quasi-periodic sensor traffic, SECON04.

[11] Tao Zheng; Radhakrishnan, S.; Sarangan, V; PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, Proceedings. 19th IEEE International on Parallel and Distributed Processing Symposium, April 2005. IPDPS2005.

[12] Tijs v. Dam, Koen Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, Proceedings of the 1st international conference on embedded networked sensor systems 2003.

[13] Van Greunen, Dragan Petrović, Alvise Bonivento, Jan Rabaey, et al., “Adaptive Sleep Discipline for Energy Conservation and Robustness in Dense Sensor Network” ICC 2004

[14] Wei Ye, John Heidemann, Deborah Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking (TON)**,** Volume 12 Issue 3, June 2004.

[15] Xiaofei Wang, Toby Berger, Self-Organizing Redundancy-Cellular Architecture for Wireless Sensor network, WCNC2005

[16] Yew-Fei Wong, Lek-Heng Ngoh, Wai-Choong Wong, An adaptive wakeup scheme to support fast routing in sensor networks, Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, 2005

[17] Yuan Li, WeiYe John Heidemann, Energy and latency control in low duty cycle MAC protocols, WCNC2005

[18] Zongpeng Li, Baochun Li, Probabilistic power management for wireless ad hoc network, MONET2005