End-to-End Performance Characteristics in Energy-Aware Wireless Sensor Networks

Feng Chen Falko Dressler Armin Heindl

Computer Networks and Communication Systems
University of Erlangen-Nuremberg
91058 Erlangen, Germany
{feng.chen,dressler,armin.heindl}@informatik.uni-erlangen.de

ABSTRACT

In this paper, we study end-to-end performance characteristics of S-MAC, an energy-aware medium access control (MAC) protocol for wireless sensor networks (WSN). WSNs are composed of battery-driven communication entities. In order to complete a given task, all sensor nodes, which are deployed in an ad-hoc fashion, have to collaborate by exchanging and forwarding measurement data. S-MAC was proposed to provide an energy-efficient MAC protocol for wireless sensor networks. It provides similar features like traditional protocols such as IEEE 802.11. Additionally, new mechanisms were added to allow self-configuration and more energy conservation. The most prominent novel feature is adaptive listening. In this paper, we provide further insight into the behavior of adaptive listening. As opposed to transient experiments on hardware, we study an S-MAC implementation – adapted from an existing one – in the network simulator ns-2. We evaluate the steady-state S-MAC performance in multiple simulations, where we apply simulation control techniques. End-to-end performance characteristics, like end-to-end delays and their jitter, are investigated under different load conditions on a multi-hop network. From our results, we conclude that end-to-end behavior in these networks may be very sensitive to load and other conditions and that careful measurements have to be made to ensure a good operating point for WSNs.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Performance attributes; C.2.m [**Computer-Communication Networks**]: Miscellaneous; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

General Terms

Performance

Keywords

Wireless sensor network (WSN), end-to-end performance, energy efficiency, medium access control (MAC)

1. INTRODUCTION

Wireless sensor networks (WSNs) have become a major research domain during the last decade [9]. Compared with other ad-hoc networks, WSNs are characterized by quite different parameters. Nodes have much fewer resources (energy, processing speed, storage), while they are exposed to requirements such as increased lifetimes and difficult environmental conditions [8]. While many aspects have already been investigated [6], we concentrate on the performance characteristics of media access control (MAC) protocols, in particular on the S-MAC protocol.

S-MAC [14] has been proposed by the SCADDS group at USC/ISI [4] as an energy-efficient MAC protocol especially designed for wireless sensor networks (letter S stands for sensor networks). To achieve one of the primary goals in sensor networks, namely energy conservation, S-MAC combines several techniques, like periodic listen and sleep, overhearing avoidance, message passing, etc., the basic ideas of which are essentially known from other protocols [13], like the IEEE 802.11 protocol for wireless LANs. Until now, especially the power-optimized scheduling of multiple nodes has been analyzed [12]. As the most prominent novel feature, adaptive listening has been introduced in S-MAC.

With respect to energy saving, the scheme of periodic listen and sleep plays the key role. However, it achieves good energy performance only at the cost of increased latency and lowered bandwidth utilization in multi-hop transmissions. Adaptive listening mitigates these side-effects. Experimental results on hardware [15] have shown the efficiency of adaptive listening in reducing latency and increasing throughput.

In this paper, we add more insight into the performance characteristics due to adaptive listening. As opposed to transient experiments on hardware, we study an S-MAC implementation in the network simulator ns-2 [2], whose output we process to investigate the stationary performance of the S-MAC protocol. To produce statistically significant results, simulation control techniques based on independent replications are applied to all our simulations. Furthermore, we look at an extended set of performance measures with an emphasis on end-to-end characteristics, including for example the jitter of end-to-end delays. Furthermore, the impact of both deterministic and highly variable traffic sources is ex-

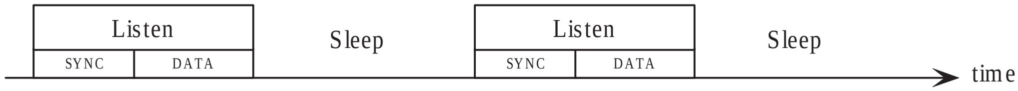

Figure 1: Periodic listen and sleep

amined in our simulations. Simulation results are compared for three different S-MAC modes of operation. Generally, our goal is to reveal, under different traffic situations, how effectively S-MAC has achieved energy conservation with the scheme of periodic sleep and improved latency in multi-hop transmission with adaptive listening.

The contributions of this paper may be summarized as follows.

1. Different versions of available S-MAC implementations in ns-2.28, which covered selective features of the protocol, have been merged to allow a comprehensive study of the S-MAC protocol.

2. Steady-state S-MAC performance is evaluated via simulation in ns-2, where we apply simulation control techniques on the fly to obtain statistically significant results.

3. End-to-end performance characteristics, like delays, are investigated under deterministic or variable load conditions on a multi-hop network.

The remainder of the paper is organized as follows: Section 2 recalls the main features of S-MAC that we consider in the simulations presented here, among which adaptive listening will be illustrated in detail according to its implementation in ns-2.28. Section 3 describes the settings, in which we simulated S-MAC with ns-2. In Section 4, three simulation scenarios and corresponding results are presented. Section 5 concludes the paper.

2. THE S-MAC PROTOCOL

To achieve energy conservation and other design goals for wireless sensor networks, several techniques have been proposed in S-MAC. In this section, we mainly discuss the S-MAC features, which are closely related to our simulations: periodic listen/sleep, overhearing avoidance and adaptive listening.

2.1 Periodic Listen and Sleep and Overhearing Avoidance

In traditional wireless communication, idle listening is identified as the dominant factor of energy waste. Periodic listen and sleep is proposed as the primary feature in S-MAC to reduce energy consumption in idle listening. The basic idea is to make each node follow a periodic listen and sleep schedule, as shown in Figure 1. A complete listen and sleep cycle is called a frame. In S-MAC, the length of the listen period is usually fixed, while the length of sleep period can be changed according to different applications, i.e., dependent on the traffic load. S-MAC defines an adjustable parameter called *duty cycle*, which is the ratio of the listen period to the frame length.

Each node has to first choose a schedule to follow before starting to work. To make neighboring nodes synchronize (follow the same schedule) and avoid long-term clock drift, S-MAC broadcasts a SYNC packet [15] periodically, e.g. every ten frames. To prevent the interference between SYNC packets and data packets, the listen period is further divided into two parts, the SYNC period and the DATA period. Generally, exchanging SYNC packets between neighboring nodes takes place during the SYNC period, while transmitting data packets starts with the DATA period (if adaptive listening is not applied).

Sending packets follows a similar contention mechanism as in the IEEE 802.11 Distributed Coordination Function (DCF, [5]). Both the SYNC and the DATA period contain a contention window with a number of slots. S-MAC has each node perform both physical and virtual carrier sense before participating in the contention. In fact, S-MAC extends the virtual carrier sense mechanism in IEEE 802.11, i.e., the network allocation vector (NAV), by considering an additional NAV, the so-called neighbor NAV. Descriptions of these NAV mechanisms can be found in [5] and [7], respectively. Thus even stricter conditions are imposed before the channel is declared idle. In case the backoff procedure has to be invoked (the same conditions as in IEEE 802.11), the slotted backoff interval is chosen randomly (according to a discrete-uniform distribution) within a fixed contention window, i.e., backoff procedure is not binary exponential. When the channel may be accessed, broadcast packets are sent directly without an acknowledgment, while unicast packets follow the RTS/CTS/DATA/ACK handshake sequence. Like in the IEEE 802.11 DCF, S-MAC uses the RTS/CTS pair to reserve the medium for the whole transmission and records the remaining time of the ongoing handshake for each unicast packet. Actually, the neighbor NAV is used to record these times also, when a node is involved in the handshake as sender or receiver (e.g., when receiving an RTS).

Overhearing occurs, when a node receives packets that are destined for other nodes, which is an obvious waste of energy. To avoid overhearing long data packets and following ACKs, an S-MAC node will go to sleep after it overhears an RTS or CTS packet. Even more, no matter what packet the node has overheard, it updates its NAV by the duration time in the overheard packet before it goes to sleep. Then the overhearing node wakes up when its NAV becomes zero. Receiving corrupted packets causes a special treatment of the NAV counters.

For transmitting data packets, S-MAC will check, if it has a buffered data packet to send only at the start of each DATA period (if adaptive listening is not applied). Only these packets may be transmitted in this DATA period. In the default settings of S-MAC, the DATA period is chosen so short that each node can either send or receive only one data packet in a frame length.

In a multi-hop network, some nodes may follow more than one schedule at the same time, each of which is realized with a timer. The schedule timer expires at the end of each of the three periods (SYNC, DATA, SLEEP) and reschedules itself for the next period. Every time when it expires (called a check point), S-MAC decides what to do in the coming

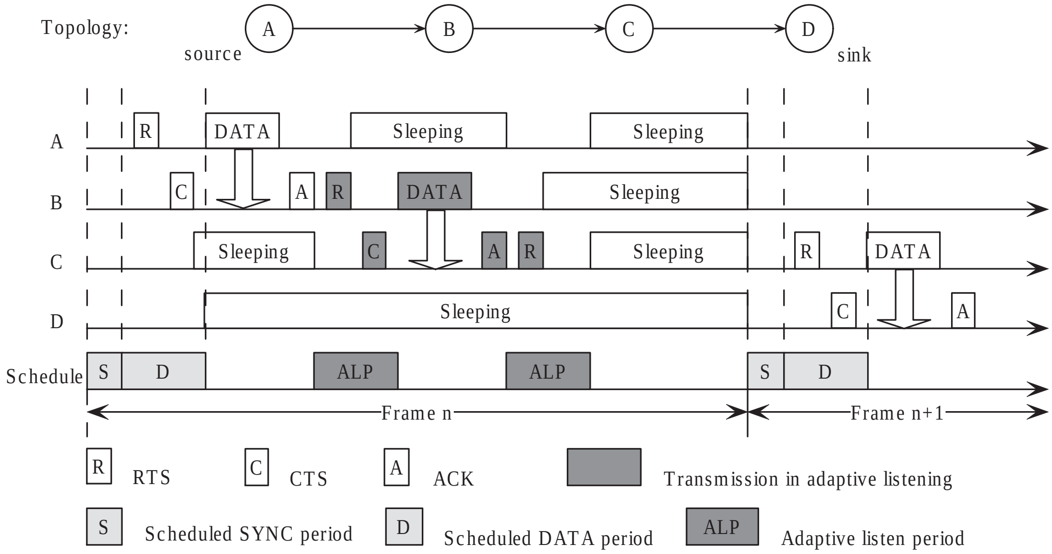

Figure 2: Transmitting a data packet through a three-hop network with adaptive listening

period. The actions to be executed during a certain period depend on many factors, including the current MAC layer state, radio state, neighbors' state, etc. For example, when the sleep period comes, S-MAC will not go to sleep if it is in the middle of an ongoing transmission or for other reasons [7]. The reader is referred to the S-MAC source code in ns-2.28 for detailed information about when S-MAC goes to sleep and wakes up.

We also do not describe message passing here, which slightly differs from fragmentation bursts in IEEE 802.11. Message passing is not considered in the simulation experiments of this paper.

2.2 Adaptive Listening

The periodic listen and sleep mechanism reduces energy consumption, but increases latency in multi-hop transmissions. The reason is that, in S-MAC without adaptive listening, each node commonly has strictly one chance to either send or receive a particular data packet in a frame. Therefore, each data packet can jump only one hop in a frame and will have a potential delay at each hop. To overcome this disadvantage, S-MAC introduces adaptive listening to improve latency. The basic idea is that at the end of one unicast transmission (normally within the scheduled sleep period), all nodes involved in a transmission, including the sender, receiver and those that overhear the transmission, will obtain an extra DATA period, called adaptive listening period (ALP). During the ALP, the node can start carrier sensing or receive packets, just the same as during the scheduled data period.

We use the example in Figure 2 to illustrate how adaptive listening reduces multi-hop latency. Suppose that four nodes are put in a row to form a three-hop network and each node can hear only its immediate neighbors. The source node A sends a data packet to the sink node D through B and C. We assume all nodes share the same schedule. Node A initiates the transmission for the scheduled DATA period and sends the data packet to B using the scheduled sleep period. During this transmission, C goes to sleep after it overhears the CTS packets from B and wakes up when the transmission is over. D is not aware of the transmission between A and B and goes to sleep when the scheduled sleep period comes. At the end of the transmission from A to B, adaptive listening is triggered on A, B and C in order to give each of them another transmission chance. Each of them checks if there is a data packet waiting in the buffer. In this case only B has the data packet to send. Therefore, it starts carrier sensing and then passes the data packet to C. During this period, A overhears the RTS packet sent by B and goes to sleep. When the transmission between B and C is over, the adaptive listening will be triggered again on A, B and C. However, this time, C fails to transmit the data packet to D and encounters a CTS timeout, because D is sleeping. For A, it has nothing to send and hears nothing. Therefore, at the end of the adaptive listening period, both A and C go to sleep and do not wake up again until the next SYNC period comes[1]. For B, it goes to sleep after it overhears the RTS packet from C. When B's NAV becomes zero, a third adaptive listening will not be triggered on B, because the remaining sleep period of the current frame is shorter than an ALP. Therefore, B keeps sleeping to the end of the current scheduled sleep period. Otherwise, i.e., if another ALP can be accommodated in the current sleep time, there would be another ALP for node B only. When the next scheduled DATA period comes, the data packet stuck at C will be transmitted to D.

We can see from the above discussion that, with adaptive listening, one data packet can jump two hops in a frame time (but usually not more), which can significantly reduce the overall latency in multi-hop transmissions.

3. SIMULATING S-MAC WITH NS-2

S-MAC has been implemented in both TinyOS on the Mote platform and the network simulator ns-2. Ye et al. [15] presented experimental results of S-MAC performance on

[1]Here, we assume that the CTS timeout at node C expires before the ALP.

Table 1: Modified S-MAC Parameters

Size of contention window for SYNC	15 slots
Size of contention window for DATA	31 slots
Length of data frame	100 bytes

Mica Motes. Here, we perform steady-state simulations with ns-2.28 to evaluate its long-run performance under different traffic conditions.

On the one hand, although an ns-2 model of the S-MAC protocol is provided by the developers of S-MAC, we found that several modifications were necessary to tune the model to our needs (with respect to S-MAC operation modes, routing protocol and statistics collection). On the other hand, we added a simulation control to ns-2 based on independent replications by means of scripts. The following subsections highlight these issues.

3.1 Source Code Modifications

The S-MAC features presented in [15] have been implemented in ns-2.28. In the source code of S-MAC, we found and fixed some bugs, which have been outlined in [7]. To show the effectiveness of the S-MAC features mentioned above, we configured the same source code to make S-MAC run under one of the following three modes [7].

1. Mode 1: No periodic sleep. Nodes go to sleep only for overhearing avoidance.
2. Mode 2: 10% duty cycle without adaptive listening.
3. Mode 3: 10% duty cycle with adaptive listening.

Our modifications allowed us to produce comparable results for all three modes.

Since the synchronization procedure is not in the focus of our interest, we modified the source code to make all nodes choose the same schedule with the initial listen period [7]. In this way, there will be always one schedule on the network during the simulation. To make parameter settings in our simulations consistent with those in the hardware experiments of [15], we modified the default values of some S-MAC parameters in the source code as listed in Table 1.

3.2 Topology and Node Configuration

The topology that we set up for our simulation model is a ten-hop linear network with sources at the first and sinks at the last node, as shown in Figure 3. Since the radio transmission range in ns-2 is 250 meters in the case of the default settings, neighboring nodes are placed 200 meters apart to make each node hear only its immediate neighbors.

The simulated nodes are configured using the parameters listed in Table 2. Note that instead of using one of those that the distribution of ns-2 provides, we employ a third-party routing agent called NOAH (NO Ad-Hoc Routing Agent)[3], which supports static multi-hop routing and produces no routing control packets. Our purpose is to eliminate the influence of routing protocols on our measurements of S-MAC performance.

Since we investigate the end-to-end performance of S-MAC, a pair of UDP and null agents are attached to the source node and the sink node respectively to set up a end-to-end logical link. In different simulation scenarios to be described in section 4, a certain type of traffic source will be connected to the UDP agent to function as a packet generator at the application layer of the source node. The intermediate nodes generate no data packets and only forward the data packets to next hop.

Table 2: Node Configuration Parameters

Channel type	WirelessChannel(Lossless)
Radio propagation model	TwoRayGround
Antenna model	OmniAntenna
Network interface type	WirelessPhy
MAC type	SMAC
Interface queue type	PriQueue
Buffer size of IFq	50
Routing protocol	NOAH

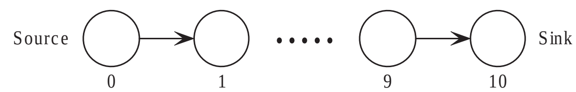

Figure 3: Topology: ten-hop linear network with one source and one sink

3.3 Definition of Performance Measures

For our study, four performance measures are considered and appropriate statistics are collected during the simulation runs:

1. Mean energy consumption per payload byte: the sum of energy consumptions measured at all the nodes divided by the total payload bytes received by the sink node.
2. Mean end-to-end delay: the sum of end-to-end delays for all data packets received by the sink node divided by the number of received packets.
3. End-to-end goodput: the total payload bytes received by the sink node divided by the time from the first packet generated at the source node to the last one received by the sink node.
4. Coefficient of variation for end-to-end delays: the standard deviation of end-to-end delay samples divided by the sample mean of end-to-end delays.

With respect to energy consumption, we only consider the energy consumed by radios. In ns-2.28, S-MAC defines a variable for tracing the change of the radio state. There are four radio states: idle, transmitting, receiving and sleeping. To measure the energy consumption at each node, we first modify the S-MAC source code to record the radio state and the current simulation time after each change of the radio state. While ns-2 is running, all radio trace information is printed out and exported to a separate trace file. Finally, a Perl script is used to extract the obtained radio trace file and compute the continuous-time statistic of the average energy consumption. The radio power values used to compute energy consumption in idle, transmitting, receiving,

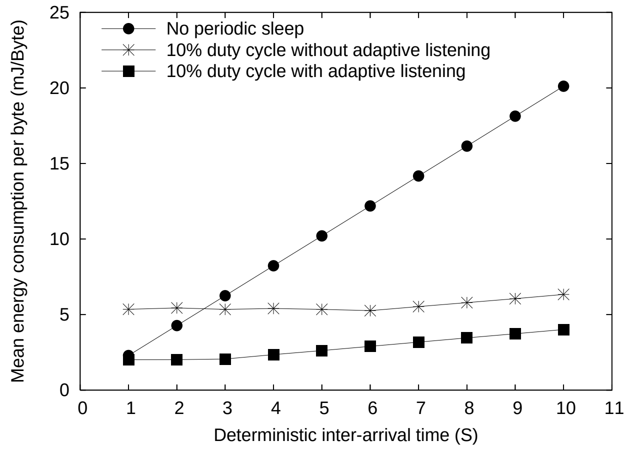

Figure 4: Mean energy consumption per payload byte under CBR traffic source

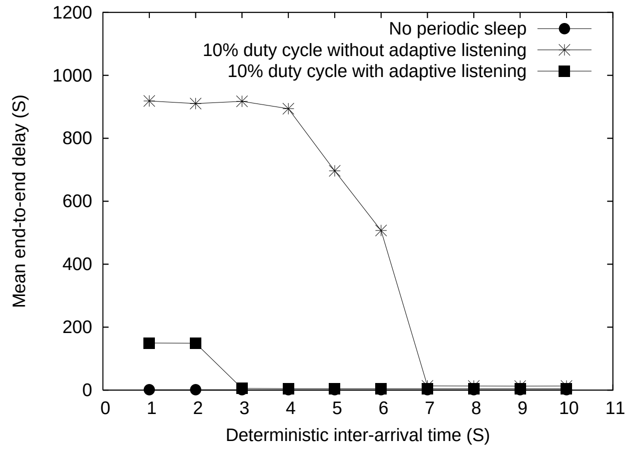

Figure 5: Mean end-to-end delay under CBR traffic source

and sleeping state are 14.4 mW, 36 mW, 14.4 mW, and 15 μW, respectively, in accordance with the RFM TR3000 radio transceiver [1] on Mica Motes.

3.4 Simulation Control in NS-2

By default, the network simulator ns-2 has no provisions to control a simulation experiment in order to produce statistically significant results, e.g., by means of independent replications. For our steady-state simulations, we implemented such simulation control techniques using Perl and Bash scripts [7]:

- Transient period detection: in order to discard initial observations biased by the initial state of the system, we implemented a rule of thumb [11], which detects the end of the transient period, if the observations have crossed the respective sample mean 25 times. In our experiments, it was sufficient to apply this rule of thumb to the measure *mean end-to-end delay* only.

- Independent replications: a confidence interval is constructed for each measure with the specified precision to control the number of replications required for a single simulation experiment. In each replication, the traffic agent at the source code keeps generating packets. The simulation time is chosen long enough to guarantee that at least 8000 packets can be received by the sink node. For the simulations presented in this paper, a relative error of 10% and a confidence level of 99% were chosen, which lead to the maximum number of 36 replications for the most variable input traffic. The confidence intervals were generally too small to be discernible in the figures of the next section, so we omitted them therein.

4. SIMULATION RESULTS

This section presents the simulation scenarios and the corresponding results. We attach two types of well-known traffic sources in ns-2, the constant bit rate (CBR) source and the exponential On/Off source, to the source node and consider three scenarios. The traffic sources generate packets with a fixed size (80 bytes, to which 20 bytes of headers will be added). Message passing [15] (i.e., fragmentation) is not considered in the experiments.

4.1 Single CBR Traffic Source

We vary the fixed packet inter-arrival time of the CBR source from 1 to 10 s. Figure 4 shows the mean energy consumed in transmitting a single byte under different traffic loads. In accordance with [14], we plotted the inter-arrival time on the x-axis, which means that the traffic load decreases with increasing values of the x-parameter. The scheme of periodic listen and sleep shows its great power in saving energy, especially when the traffic load is light, because it largely reduces the energy consumed in idle listening. Under heavy traffic loads, S-MAC with adaptive listening consumes only half the energy that is consumed by the one without adaptive listening. We see the reason for this in that adaptive listening keeps awake only a selected group of nodes for the ALP so that the next sender and receiver can initiate their handshake almost without contention. This results in much fewer collisions and thus considerable energy savings for useless transmissions. This is another favorable effect of adaptive listening in addition to the reduced latency in multi-hop transmissions with periodic sleep schemes.

Figure 5 shows this latency efficiency of adaptive listening for the end-to-end delays. The ten-hop network with a large queue size (50 for each node) results in a huge saturation delay under very heavy traffic loads. For S-MAC without adaptive listening, the delay exits saturation and starts to drop rapidly when the inter-arrival time is larger than 4s. In S-MAC, two nodes within the range of two hops will interfere with each other. When the packet inter-arrival time is smaller than the mean delay for passing a data packet over three hops, interferences will happen frequently, which in the long run leads to serious network traffic jam and large packet delays. Another simulation presented in [14] has shown that the mean delay for passing a data packet over three hops is very close to 4s, which proves that the turning point at 4s is reasonable. For S-MAC with adaptive listening, its turning point is only half of that in S-MAC without adaptive listening and it has much lower saturation delay. The reason is that adaptive listening can make data packets jump two hops in a frame, which significantly reduces queue delays.

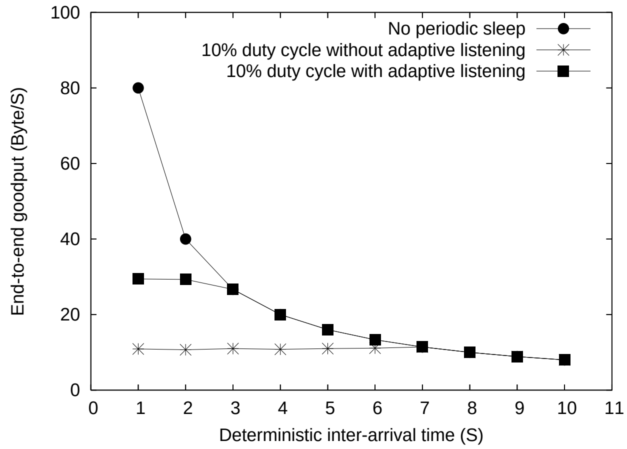

Figure 6: Mean end-to-end goodput under CBR traffic source

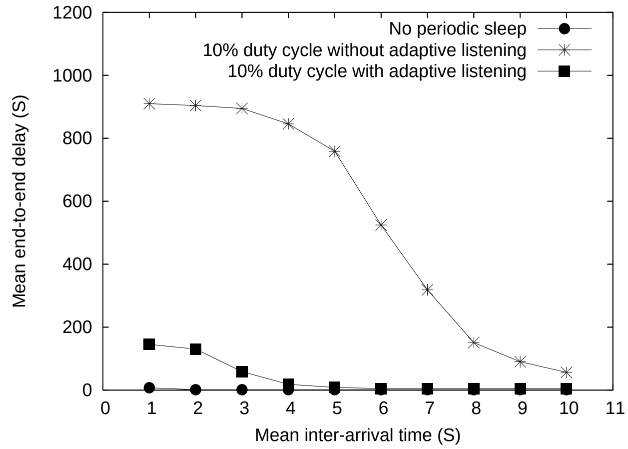

Figure 7: Mean end-to-end delay under On/Off traffic source

Figure 6 shows the measured end-to-end goodput. We can see from it that with periodic sleep, the achievable goodput of the multi-hop network is considerably reduced. Adaptive listening improves the bandwidth utilization by reducing the latency, since sleep periods may be bridged by ALPs.

From the above three results, we can see that periodic sleep helps S-MAC achieve very good energy performance under light traffic loads. Under heavy traffic load, S-MAC with adaptive listening greatly improves the problems of large latency and low bandwidth utilization that periodic sleep has brought. However, the mean end-to-end delays are still huge compared with mode 1 under light traffic loads (around 1.4s, 13s, 4.7s, in mode 1, 2, 3, respectively). In latency-sensitive applications, we can solve this problem by adjusting the duty cycle of S-MAC to a relatively large value, i.e., with longer listen periods relative to the frame length.

Increasing the duty cycle may of course be the measure of choice to avoid the sharp increase in the end-to-end delays with increasing load (see Figure 5). To this end, one would have to be able to determine the turning point, i.e., the critical load, in advance. This, however, may be difficult, since the critical load depends on various factors. The next experiment shows the impact of more variable traffic on this turning point.

4.2 Single On/Off Traffic Source

In this scenario, we investigate the performance of S-MAC utilizing an exponential On/Off traffic source. To make the On/Off source generate a traffic flow with an identical mean inter-arrival time as in the last scenario, we let the mean time in the "On" state equal that in "Off" state and set it ten times longer than the packet inter-arrival time during "On" period, which varies from 0.5s to 5s. In other words, if the CBR traffic has an inter-arrival time twice as large as the one during the "On" period of the On/Off traffic, both sources will generate the traffic with the same mean arrival rate. However, the On/Off source will exhibit a higher variability.

Comparing the results under two different traffic sources, we find that the obvious difference is that the curve of end-to-end delays is smoothed by the exponential behavior of the On/Off source, which is shown in Figure 7. The smoothing effect is not so noticeable for the other performance measures. Because of only minor differences for the corresponding figures in the case of a single CBR traffic source, we omit the figures for energy consumption and goodput here.

4.3 Combined Traffic Sources

In this scenario, one CBR and one On/Off traffic source generate packets simultaneously at the source node. The purpose is to show how in different S-MAC modes a CBR source is influenced by a noise source with a burst nature (On/Off). We fix the packet inter-arrival time of the CBR source to 7s, which is a moderate traffic load (below the critical load or above the turning point in Figure 5), and vary the interval-times during on periods at the On/Off source from 0.25 to 10s. Figure 8 shows the mean end-to-end delays of the CBR packets. The delay in mode 2 is much larger than that of the other two modes and keeps at a high level. The reason is that the fixed inter-arrival time for the CBR source with 7s is right of the turning point, from which the delay starts increasing rapidly with slightly higher load (see Figure 5). Therefore, even a small amount of noise will cause blocked traffic and long queue delays. As for S-MAC with adaptive listening, the delay stays at a low level until the inter-arrival time of the On/Off source is decreased to 3.5s. This value leads to an overall mean inter-arrival time of about 3.5s, which is close to 3s, the turning point for S-MAC with adaptive listening under one CBR traffic source shown in Figure 5. The slight deviation is caused by the smoothing effect of the exponential On/Off source.

The other measure of this scenario is the coefficient of variation (CV) for end-to-end delays, as shown in Figure 9. From the figure, we can see two peaks in mode 1 and mode 3 respectively, which indicates a high jitter in delays. The results of the CV show us that when the traffic load is moderate, S-MAC with adaptive listening may show a bad jitter performance in regions of the turning point. However, when the traffic load is very heavy, S-MAC with adaptive listening will show much better jitter performance than S-MAC without periodic sleep. As for S-MAC without adaptive listening, which has a smooth CV curve, it is insensitive to the noise and has much better jitter performance than the other two modes. However, this advantage is achieved at the cost of high end-to-end delays and low bandwidth utilization.

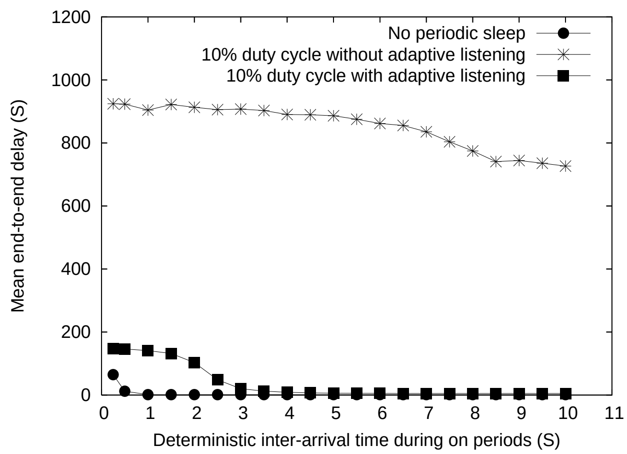

Figure 8: Mean end-to-end delay under the combination of CBR and On/Off traffic sources

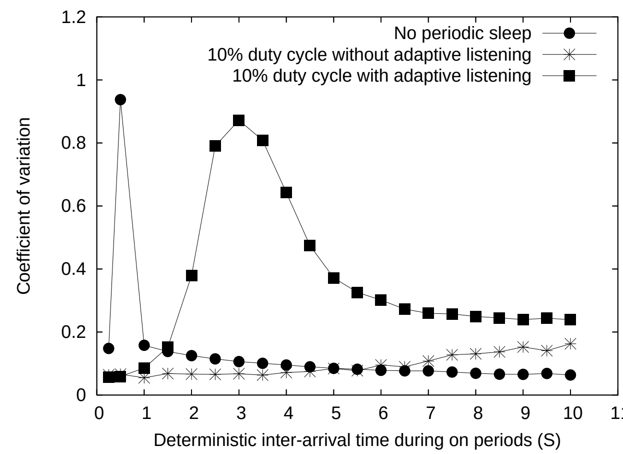

Figure 9: Coefficient of variation for end-to-end delays under the combination of CBR and On/Off traffic sources

5. CONCLUSION

This paper presents a simulation study of S-MAC with ns-2. Our simulation results reveal the trade-off between energy consumption and latency (goodput) in stationary S-MAC operation. When the traffic load is not heavy, S-MAC shows its great advantages in saving energy. However, with increasing loads, the drawbacks of a periodic sleep scheme are exposed, although adaptive listening can compensate them to a certain extent. Furthermore, we conclude from our experiments that – for a proper operation of an S-MAC sensor network – it is very important to identify the critical load beyond which the mean end-to-end delays sharply increase. This turning point may of course depend on many factors, like the density and topology of sensor nodes. In future work, heuristic rules should be found to take appropriate countermeasures, like adjusting the duty cycle, to cope with higher load. However, in both energy-aware and end-to-end delay-constrained WSN applications, the full synchronized wakeup pattern like S-MAC may not be the best choice. Since the effort contributed by improving the MAC layer protocols is limited, recent researches are focused on the cross-layer design approach to achieve better trade-offs between energy and end-to-end delays. Examples include fast path algorithm in [12] and multi-parent schemes in [10].

6. REFERENCES

[1] ASH transceiver TR3000 data sheet. [online]. available: http://www.rfm.com/.

[2] The network simulator - ns-2 homepage. [online]. available: http://www.isi.edu/nsnam/ns/.

[3] NO Ad-Hoc Routing Agent - NOAH homepage. [online]. available: http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/.

[4] SCADDS project at USC/ISI homepage. [online]. available: http://www.isi.edu/scadds/.

[5] Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE std. 802.11-1999 edition.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–116, 2002.

[7] F. Chen. *Simulation of Wireless Sensor Nodes Using S-MAC*. Master's thesis, University of Erlangen-Nuremberg, 2005.

[8] C.-Y. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[9] D. Culler, D. Estrin, and M. B. Srivastava. Overview of sensor networks. *Computer*, 37(8):41–49, 2004.

[10] A. Keshavarzian, H. Lee, and L. Venkatraman. Wakeup Scheduling in Wireless Sensor Networks. In *Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc 2006)*, Florence, Italy, May 2006.

[11] A. M. Law and D. M. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 1999.

[12] Y. Li, W. Ye, and J. Heidemann. Energy and latency control in low duty cycle mac protocols. Technical report isi-tr-595, USC Information Sciences Institute, August 2004.

[13] W. Ye and J. Heidemann. Medium access control in wireless sensor networks. Technical report isi-tr-580, USC Information Sciences institute, October 2003.

[14] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1567–1576, New York, NY, USA, 2002.

[15] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 12(3):493–506, 2004.