Design and Implementation of a Look-Ahead Scheduling MAC Protocol for Wireless Sensor Networks

Jaesub Kim Kyu Ho Park
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
373-1 Guseong-dong, Yuseong-gu, 305-701 Daejeon, Korea
jskim@core.kaist.ac.kr, kpark@ee.kaist.ac.kr

ABSTRACT

Energy is the most important resource in wireless sensor networks. The wireless interface of sensor node consumes most of the energy and therefore many MAC protocols are adopting the periodic listen-and-sleep scheme. However, the periodic listen-and-sleep approach results in high latency and low throughput. For low latency in multi-hop forwarding without sacrificing energy efficiency, we designed a look-ahead scheduling MAC (LAS-MAC) protocol, which also works on the periodic listen-and-sleep scheme, but it reserves multi-hop packet forwarding schedules across multi-hop nodes during the listen period and forwards data packets by awaking the nodes with the reserved schedules during the sleep period. For high throughput of LAS-MAC, in addition, we add the throughput enhancement mechanism which clones the current multi-hop forwarding schedules for the subsequent data packet forwarding. Our experimental results on the MICA2 platform show that LAS-MAC achieves lower latency, higher throughput, and higher energy efficiency than 802.11-like MAC without sleeping.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication-MAC protocol*

General Terms

Design, Performance

Keywords

Wireless Sensor Networks, MAC, LAS-MAC, Implementation, Energy Efficiency, Low Latency, High Throughput

1. INTRODUCTION

Wireless Sensor Networks (WSNs) enable a wide range of emerging applications such as environment monitoring, mobile target tracking, smart space, and ubiquitous computing. Sensor nodes are operated by batteries and therefore energy is the most important resource in WSNs. The major source of energy waste in a sensor node is idle listening of the wireless interface, which consumes considerable energy doing nothing.

In order to reduce idle listening, many MAC protocols such as S(Sensor)-MAC [9] adopted the periodic listen-and-sleep approach, but it caused long latency and low throughput problems in multi-hop packet forwarding. Let us assume that all sensor nodes are synchronized and listen and sleep together. If two nodes (a sender and a receiver) acquire the shared wireless channel and transmit a packet during the listen period, the other nodes around them cannot help overhearing or sleeping during their transmission. After two node's packet transmission, the listen period ends and therefore the nodes with any packet to transmit (including the receiver) must wait until the next listen period. It causes inefficiency in using the shared wireless channel, thus leading long latency and low throughput, especially in multi-hop packet forwarding.

For low latency in multi-hop forwarding, T(Timeout)-MAC [8] sends a Future-Request-To-Send (FRTS) packet and S-MAC uses an adaptive listening scheme [10] for the next forward node to keep awake. However, they cannot make the nodes in several hops away awake because of a limited wireless transmission range; thus, they still experience the long latency. For much lower latency, we proposed a look-ahead schedule MAC (LAS-MAC) protocol [7]. LAS-MAC is also based on the periodic listen-and-sleep scheme. However, LAS-MAC uses the listen period for multi-hop forwarding schedule (channel) reservation across the multi-hop nodes, and the sleep period for execution of the reserved schedules. During the sleep period, the nodes with the reserved schedules wake up and forward the data packets for the scheduled time, and then sleep again. In this way, LAS-MAC can forward packets many hops per period, without sacrificing energy efficiency. The Routing-enhanced duty-cycle MAC (RMAC) [5] works similarly as LAS-MAC but it is simulated and has long latency when packet transmission errors occur, because RMAC simply resumes the packet transmission in the next listen period.

In this paper, we present design and real implementation issues of LAS-MAC. In the LAS-MAC implementation, we add a throughput enhancement mechanism which allows the data packet transmission to reserve the next forwarding schedules of the subsequent data packet, by utilizing the cross-layer interface queue (IFQ) information. If the subsequent data packet in the IFQ has the same destination as the current data packet transmission, the current packet transmission reserves forwarding schedules of the subsequent data packet. With this mechanism, LAS-MAC can forward much more data packets per period utilizing the long sleep period.

The rest of this paper is structured as follows: In Section 2, we describe our basic design and operations of LAS-MAC. In Section 3, we present the experiment environment and the performance results of LAS-MAC and other MACs. Finally, in Section 4, we

IWCMC '09, June 21-24, 2009, Leipzig, Germany.

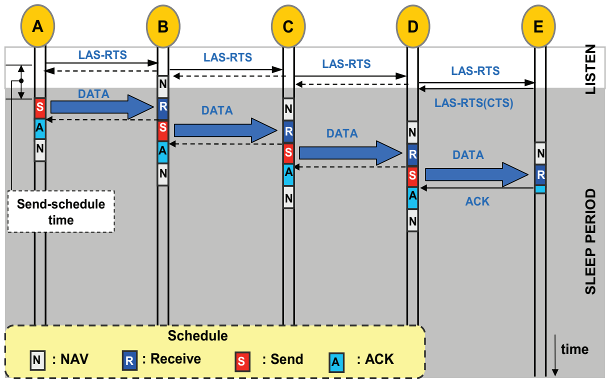


Figure 1: Overview of the LAS-MAC protocol.

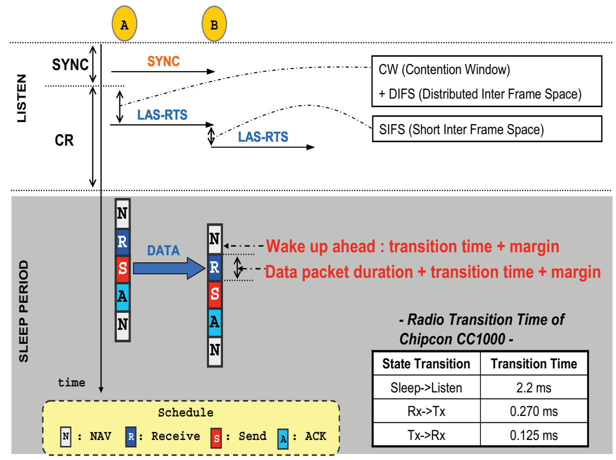


Figure 2: Implementation issues of the basic operations.

present our conclusions.

2. LOOK AHEAD SCHEDULING MAC

2.1 Basic Operations

The basic operations of LAS-MAC are shown in Fig. 1. At the beginning of the listen period, the node that has data packet to send forwards a short control packet, called a Look-Ahead Scheduling (LAS)-RTS, across multi-hop nodes for multi-hop schedule reservation. The LAS-RTS includes a destination address and a send-schedule time of the data packet in addition to the previous RTS. The destination address is used to determine the next forward node of the LAS-RTS using cross-layer routing information, and the send-schedule time represents a relative start time of the data packet transmission from the end time of the current LAS-RTS transmission, used for multi-hop schedule reservation. The duration field in the LAS-RTS only represents the duration of a data packet transmission because LAS-MAC eliminates an explicit ACK packet in multi-hop forwarding.

First, node A reserves its forwarding schedules of {S, A, N}; the duration of each schedule is equal to the duration of data packet transmission. Node A then forwards the LAS-RTS with the send-schedule time (S schedule), the final destination (node E), and the data packet duration. The next node (node B) then reserves its forwarding schedules of {N, R, S, A, N} and determines the next forward node based on the destination in the received LAS-RTS and cross-layer routing information, and forwards the LAS-RTS with a new send-schedule time. The previous node (node A) overhears node B's LAS-RTS and confirms its schedules, which substitutes for the CTS. The LAS-RTS has the functions of both the RTS and the CTS, so the LAS-RTS can be forwarded to much more hops during the short listen period. The same procedure is repeated until the listen period ends or the LAS-RTS reaches its final destination. The last node of multi-hop forwarding (node E) reserves {N, R, A} and transmits an explicit CTS to confirm the forwarding schedules of the previous node (node D). Our scheduling scheme assumes that 2-hop neighbors hardly interfere the node, but packet errors from this interference can be handled by *schedule shift* in Section 2.4.

In the sleep period, all nodes sleep (radio off) but the nodes with the reserved schedules wake up at their scheduled time and forward data packets. In N (NAV) schedule, the node sleeps because it cannot send any packet, and in A (ACK) schedule, the node receives the next node's data packet or ACK packet and regards it as an acknowledgement of its data packet transmission; the node does not have to overhear the entire data packet because the header of data packet is enough to regard it as an acknowledgement. The last node (node E) needs an explicit ACK transmission. Consequently, the nodes in the forwarding path are only awake for essential operations of packet forwarding.

LAS-MAC achieves low latency in data packet forwarding by reserving pipelined multi-hop schedules in advance, and eliminates unnecessary control packets such as the CTS and the ACK using the broadcast characteristic of wireless radio in multi-hop forwarding, resulting in an efficient multi-hop channel reservation as well as energy saving.

2.2 Implementation Issues of Basic Operations

The listen period of LAS-MAC is divided into a synchronization (SYNC) period and a channel reservation (CR) period, as shown in Fig. 2. During the SYNC period, LAS-MAC sends SYNC packets periodically and synchronizes the listen time as S-MAC [9]. At the beginning of the CR period, the nodes with data packets to send sense carrier during a contention window (CW) and a distributed inter frame space (DIFS), as in IEEE 802.11 [4], and start to forward the LAS-RTS if the channel is clear. The LAS-RTS sender reserves forwarding schedules just before the LAS-RTS transmission, and the LAS-RTS receiver reserves schedules at the reception of the LAS-RTS. There is a time gap between the reservation times of the sender and the receiver, so the receiver subtracts the LAS-RTS forwarding delay from the send-schedule time in the received LAS-RTS and uses it for schedule reservation. The LAS-RTS receiver forwards the LAS-RTS after a short inter frame space (SIFS), which is long enough time to process the packet, to reserve schedules, and to switch radio modes (from Rx. to Tx.).

Switching radio modes in a hardware requires some transition time. As an example, the table in Fig. 2 shows radio transition time of the *Chipcon CC1000* radio transceiver [1]. Especially, switching radio modes from sleep to listen takes considerable time compared to others, which causes a synchronization problem in executing multi-hop schedules because the receiver is not ready to receive at its R (Receive) schedule time. Therefore, LAS-MAC makes the radio wake up at a given time which is radio transition plus margin time ahead of its R schedule time. Similarly, the duration of each schedule also needs some extra time of radio transition plus margin in addition to data packet duration. The margin time is necessary due to the clock drift on each node and processing delay of each packet in the physical and MAC layer. With these considerations, LAS-MAC can properly work on distributed real sensor nodes.

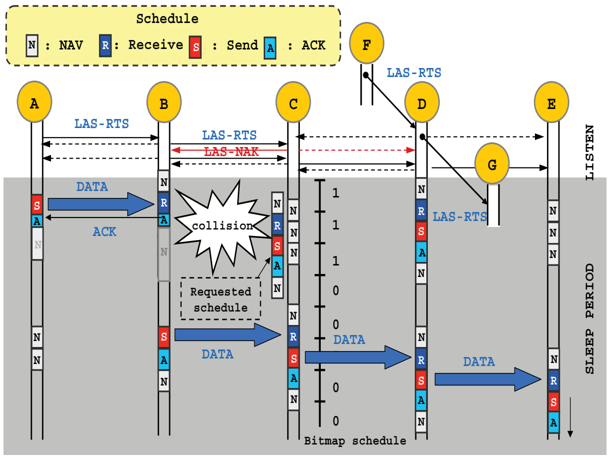


Figure 3: Schedule collision handling.

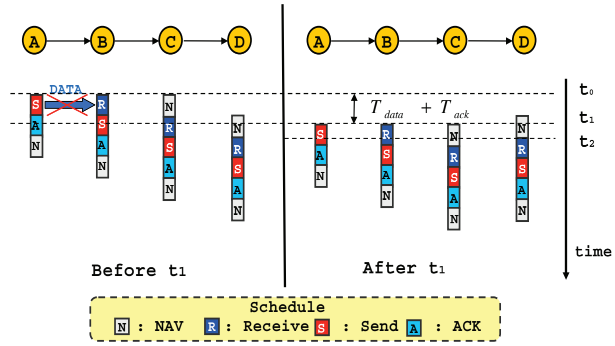


Figure 4: Data packet error handling by means of *Schedule shift* mechanism.

In general, the MAC-layer can only access the data packet after full reception of the data packet in the physical-layer. However, it only wastes energy when LAS-MAC only needs the data packet header for an ACK (at A schedule). Therefore, LAS-MAC stores the physical-layer buffer pointer if start symbol is detected at A schedule, and then obtains the data packet header from the receiver buffer pointer after a given time-out.

2.3 Schedule Collision Handling

During the channel (schedule) reservation period, some nodes may request over-lapping schedules, which is called a *schedule collision*. Fig. 3 shows the schedule collision handling mechanism.

In Fig. 3, one traffic flow goes from node A to node E, and an another cross traffic flow goes from node F to node G. First, node F of the cross traffic flow transmits the LAS-RTS to node D, and node D then reserves {N, R, S, A, N} and forwards the LAS-RTS to node G. At this point, the neighbors of node D, node C and node E, overhear this LAS-RTS and reserve {N, N, N} because they are hidden terminals when node D executes its forwarding schedules. Node A also forwards the LAS-RTS and reserves schedules, but a schedule collision occurs at node C with the prior schedules ({N, N, N}). Node C then signals a failure of schedule reservations through LAS-Negative-Acknowledgement (LAS-NAK), which is identical to the LAS-RTS except that the send-schedule time field is filled with a *bitmap schedule*. The bitmap schedule is a summary of all schedules already reserved at a node. Each bit takes charge of the fixed quantum of time (sleep period / bitmap size) and is set when a corresponding schedule exists. The reason a bitmap schedule is transmitted to node B is to make node B reschedule without additional schedule collisions. As soon as node B receives LAS-NAK, node B cancels its forwarding schedule by calling the *withdraw_multihop_ schedule*() function; thus, {N, R, S, A, N} is changed to {N, R, A}. Node B then retransmits the LAS-RTS with a newly arranged schedule ({S, A, N}) from the bitmap schedule, thereby avoiding a data packet transmission collision.

2.4 Data Packet Transmission Error Handling

LAS-MAC reserves schedules and forwards data packets in a pipelined way. Therefore, if there is a data packet transmission error in an intermediate node, all the following schedules in the forwarding path will fail.

To avoid chain errors of data packet forwarding, we suggest a *schedule shift* mechanism, as described in Fig. 4. The basic idea of the schedule shift mechanism is that the failures of pipelined schedules from a data packet transmission error can be recovered by shifting schedules in the following nodes by the same amount of time; that is, $T_{shift} = T_{data} + T_{ack}$, where T_{data} is the duration of the data packet transmission and T_{ack} is the duration of ACK packet transmission. The T_{shift} is the minimum time necessary for both the sender and the receiver to notice the failure of data packet transmission.

At the left-hand side of Fig. 4, before t_1, we can see that node A is sending a data packet to node B at t_0, but a transmission error occurs. Node B finds this error out at the end of the R schedule; it then comes back to the beginning of the R schedule and shifts its schedule by T_{shift}. Next, node C finds an error at the beginning of the R schedule because it cannot receive anything; it then comes back to the R schedule and shifts its schedule. For the schedule shift at the beginning of the R schedule, a time-out mechanism is necessary. If LAS-MAC does not receive anything for a short time-out interval from the beginning of the R schedule, it shifts its schedules and sleeps. At t_1, node A finds its data packet transmission error because it cannot receive an ACK during the T_{ack} time of the A schedule; it then shifts its schedules.

As a result, after t_1, the shifted schedules are represented at the right-hand side of Fig. 4, and the data packet is retransmitted at node A. Likewise, node D shifts its schedule at t_2 because no data is being received in the R schedule, and all the following nodes shift their schedules consecutively for the same reason. In this way, multi-hop schedules can be reused without a new expensive schedule reservation process. However, too many schedule shifts consume the extra energy because all the following nodes have to shift their schedules; thus, the number of the schedule shifts is limited. The schedule shift is canceled if the shifted schedules reach the other schedules (a schedule collision), or the end of current period; the data packet transmission then resumes in the next period.

2.5 Throughput Enhancement Mechanism

WSN generally assumes a small number of sinks for event data collection, which means that a lot of data packets use the same routing path. Exploiting this characteristic, LAS-MAC reserves the forwarding schedules of subsequent data packet in the interface queue (IFQ) as long as the subsequent data packets are destined to the same destination, by cloning the forwarding schedules of the previous data packet transmission. To clone the schedules, LAS-MAC adds schedule information (a send-schedule time) to the header of the data packet.

Figure 5 shows schedule reservation in data packet transmission. Before forwarding the first data packet, node A finds subsequent

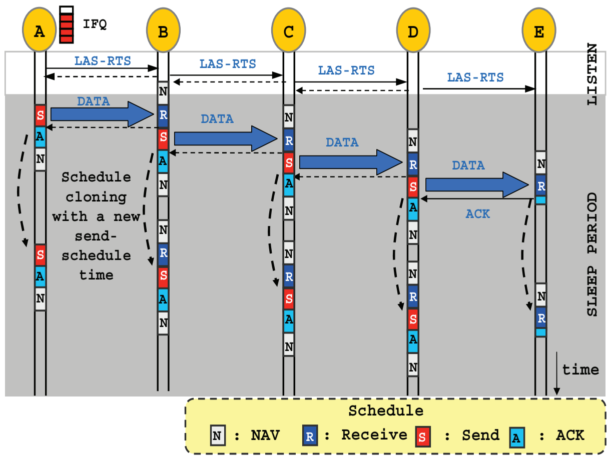


Figure 5: Schedule reservation in data packet transmission.

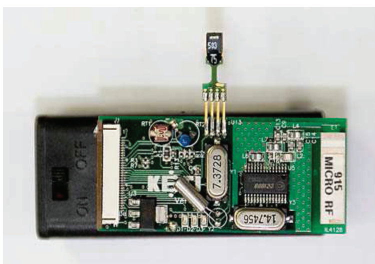


Figure 6: The Tip30 (Mica2 platform).

data is also bound to the same destination. Then, node A reserves schedules for the subsequent data packet by cloning the first data packet's schedules, and forwards the first data packet to node B with subsequent data's schedule information. Node B then clones its previous forwarding schedules based on the received schedule information and forwards its data packet with its subsequent data packet's schedule information. This procedure is repeated to node E, and repeated as long as data packets in IFQ are destined to the same destination. In this way, LAS-MAC can forward a large amount of data packets in one period without an explicit schedule reservation in the listen period, thus increasing throughput.

3. PERFORMANCE EVALUATION

The primary goal of our experimentation is to demonstrate the performance of LAS-MAC and to compare LAS-MAC with other MAC protocols such as S-MAC [9], S-MAC with adaptive listening scheme (S-MAC-ADAPT) [10], and 802.11-like MAC (802.11-LIKE) which works like S-MAC without sleeping. T-MAC [8] is not included in the performance evaluation because S-MAC-ADAPT is similar to T-MAC in that the listen period can be adaptively extended by overhearing the transmissions of the neighbors.

3.1 Experimental Environment

We use the *Tip30* which is manufactured at the KETI (Korea Electronics Technology Institute) and based on MICA2 platform [2], as shown in Fig. 6. The hardware specification of the *Tip30* is shown in Table 1.

The radio transceiver is *Chipcon CC1000* [1], which provides a bandwidth of 19 Kbps with Manchester encoding and four working modes: idle listening, receiving, transmitting, and sleeping. We

Sensor node type	The Tip30 (MICA2)
Microcontroller	ATMEGA128L
Program Memory	128KB
RAM	4KB
Communication	CC1000 (914-915MHz)
Sensor	Temperature, Humidity, Light
Default Power Source	Alk (2*AAA)

Table 1: The Hardware specification of the Tip30.

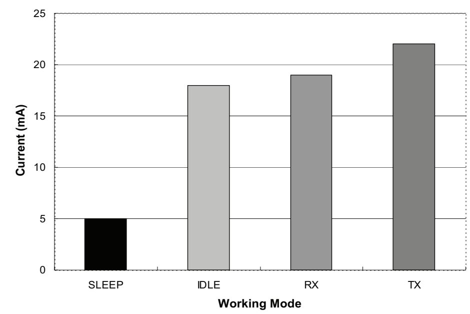


Figure 7: Power consumption of the *Tip30*.

measure the total drawing current of each working mode on the *Tip30* and show it in Fig. 7. In our experiments, we do not make CPU go to sleep during the node's sleeping mode, so there is still some drawing current. If sleeping mode of CPU is used, much less current will be drawn. We uses a SMD-type antenna (904-924MHz).

Our platform uses TinyOS (ver. 2.1) [6] and the physical stack used in S-MAC [9] [3]. The key parameters used in our experiments are shown in Table 2. The size of the LAS-RTS is 4 bytes more than the RTS since it has the additional destination (2 bytes) and send-schedule time (2 bytes) fields.

Fig. 8 shows our experimental topology, which is 7-hop indoor network with one source (one end node) and one sink (the other end node); all nodes are equally spaced (4m apart). The traffic load is changed by varying the data generation interval of the source.

We run experiments for a fixed interval of 500 seconds; the source starts to forward data packets at 120 seconds from the beginning of each experiment, so data packets are forwarded for 380 seconds.

3.2 Experimental Results

3.2.1 *End-to-End latency*

The measured end-to-end latencies of each MAC protocol are shown in Fig. 9(a). To measure the latency, the clocks of the source and the sink are synchronized, and the source puts time-stamp of each data packet into the corresponding data packet; then, the sink calculates the end-to-end latency.

LAS-MAC has a very low latency compared to other protocols, even though LAS-MAC listens and sleeps periodically. 802.11-LIKE has a lower latency than LAS-MAC above the data generation interval of 1 second, because it is always awake and handles data packets immediately, but 802.11-LIKE has a higher latency than LAS-MAC below 1 second due to the increased channel contention with the increased data generation rate; LAS-MAC experiences much lower channel contention with the pipeline-scheduled packet transmission, having a lower latency. S-MAC has a very long latency because it forwards data packets slowly (one hop per period), and thus encounters more contention with the preceding

Duty Cycle	10 %
SYNC period	47 ms
Listen period	221 ms (LAS-MAC), 143 ms (others)
Contention Window size	32 ms
SYNC pkt. size	10 bytes
RTS/CTS/ACK pkt. size	10 bytes
LAS-RTS/NAK pkt. size	14 bytes
DATA pkt. size	50 bytes
IFQ size	500 bytes (10 DATA pkts.)

Table 2: Experimental Parameters.

Figure 8: Topology used in experiments: 7-hop indoor network with one source and one sink. All nodes are 4m apart.

data packets. S-MAC delays its transmission until the next period whenever it loses during channel contentions or meets packet collisions. The latency of S-MAC-ADAPT is considerably enhanced compared to the latency of S-MAC.

3.2.2 *Throughput*

The total number of received data packets at sink during a fixed time interval of 380 seconds is shown in Fig. 9(b). We can see that the throughput of LAS-MAC is the best due to the scheduled packet transmissions. The throughput of 802.11-LIKE is lower than that of LAS-MAC because the intermediate nodes in the forwarding path contends each other to forward their data packets. The adaptive listening scheme in S-MAC-ADAPT increases the throughput of S-MAC a lot.

3.2.3 *Energy Consumption*

The percentage of each working mode during 500 seconds with the data generation interval of 2 seconds is shown in Fig. 9(c). 802.11-LIKE spends most of its time in idle listening mode, but other protocols reduce the idle listening time by periodic listening and sleeping. Although LAS-MAC and S-MAC-ADAPT have the same throughput with the data generation interval of 2 second as depicted in Fig. 9(b), LAS-MAC has a shorter idle time than S-MAC-ADAPT since LAS-MAC wakes up only when it needs. Moreover, LAS-MAC has shorter TX time than S-MAC-ADAPT since LAS-MAC eliminates the unnecessary control packet transmissions as explained in Section 2.1.

The energy consumption per data packet forwarding at the intermediate node is depicted in Fig 9(d). Among all the tested protocols, LAS-MAC consumes the least energy for a data packet forwarding in any data generation interval.

3.2.4 *Throughput with Various Radio Transmission Powers*

Throughput of LAS-MAC with various radio transmission powers and the corresponding number of data packet errors are shown in Fig. 9(e) and Fig. 9(f), respectively. The number of data packet errors of all the nodes considerably increases as the radio transmission power decreases, but throughput of LAS-MAC only decreases by less than 20% at -10db. If there is a data packet transmission error in an intermediate node, all the following schedules in the forwarding path will fail and it will decrease throughput considerably, but LAS-MAC overcomes it by means of the schedule shift mechanism.

4. CONCLUSIONS

For energy efficiency in WSNs, many MAC protocols have used the periodic listen-and-sleep scheme, which has reduced the energy consumed by idle listening considerably but has introduced long end-to-end latency and low throughput. We have presented a new listen-and-sleep MAC called LAS-MAC which is designed for low end-to-end latency and high throughput, without sacrificing energy efficiency. LAS-MAC achieves its goals by reserving the pipelined multi-hop schedules in advance during the sleep period as well as the listen period. However, the pipelined data forwarding schedule causes cascade schedule failures when the middle nodes have transmission failures; thus, LAS-MAC provides the schedule shift mechanism for it. Our experimental results show that LAS-MAC achieves low latency, high throughput, and energy efficiency even when there are some packet errors.

5. REFERENCES

[1] Chipcon CC1000 RF transceiver.

[2] Crossbow MICA2 datasheet.

[3] S-MAC source code.

[4] Wireless LAN medium access control (MAC) and physical layer (PHY) specification, 1997.

[5] S. Du, A. K. Saha, and D. B. Johnson. RMAC: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In *INFOCOM*, pages 1478–1486, 2007.

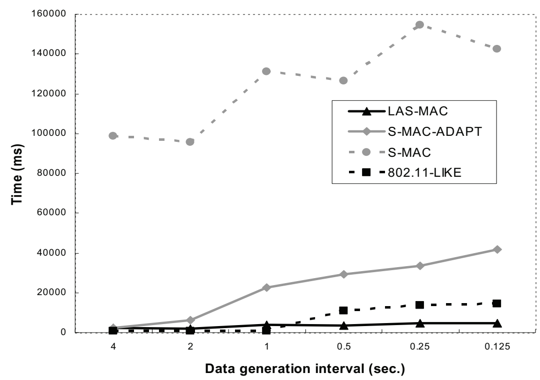
[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *ASPLOS-IX: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*, pages 93–104, New York, NY, USA, 2000. ACM Press.

[7] Jaesub Kim, Keuntae Park, Daeyeon Park. An Energy-Efficient Scheduling MAC Protocol for Wireless Sensor Networks. In *Consumer Communications and Networking Conference(CCNC) '07*, pages 655–659. IEEE, 2007.

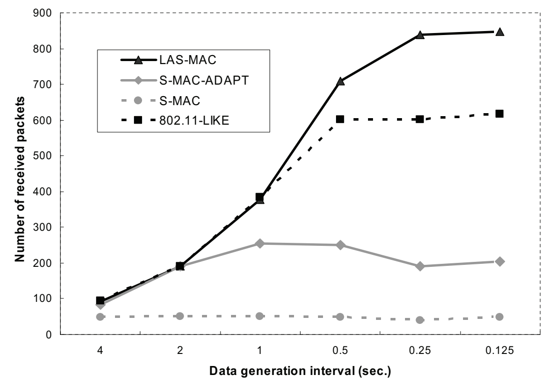
[8] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM Press.

[9] W. Ye and J. Heidemann and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Infocom '02*, pages 1567–1576. IEEE, 2002.

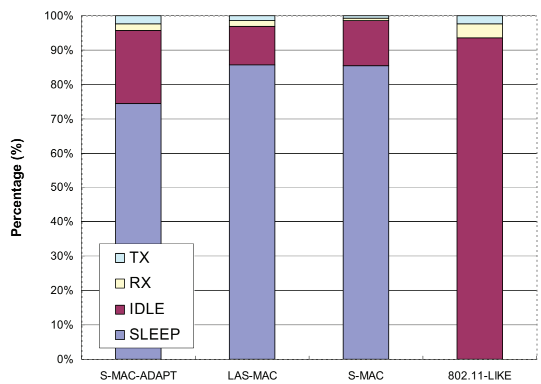
[10] Wei Ye and John Heidemann and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.



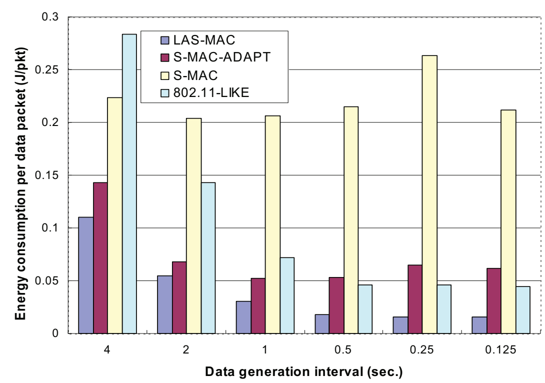
9(a) End-to-end latency.



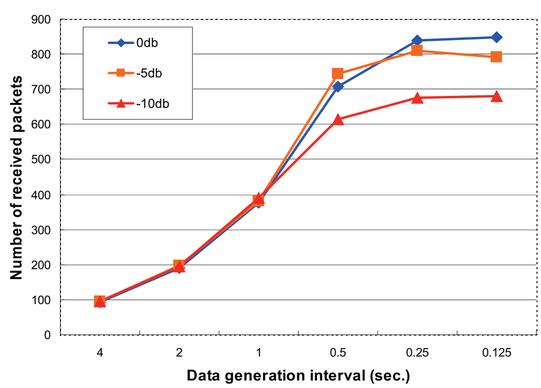
9(b) Total number of received packets at the sink (Experimental time = 380 seconds).



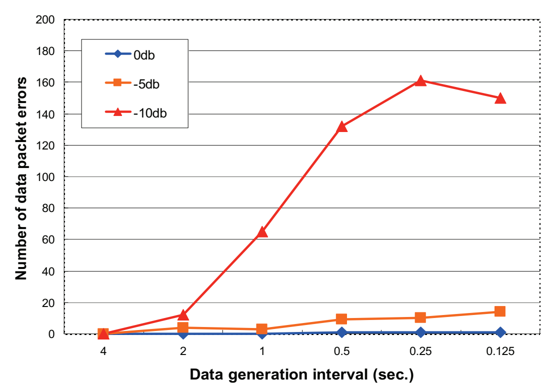
9(c) Percentage of each radio mode (Experimental time = 500 seconds, Data generation interval = 2 seconds).



9(d) Energy consumption per data packet forwarding at the intermediate node.



9(e) Throughput of LAS-MAC with various radio transmission powers (Experimental time = 380 seconds).



9(f) Number of data packet errors in LAS-MAC with various radio transmission powers (Experimental time = 380 seconds).

Figure 9: Experiment results in a 7-hop indoor network shown in Fig. 8.