A Traffic Aware, Energy Efficient MAC Protocol for Wireless Sensor Networks

Changsu Suh and Young-Bae Ko
The School of Information and Communication,
Ajou University,
Suwon, Republic of Korea
{scs, youngko}@ajou.ac.kr

Abstract— **In this paper, we focus on the problem of designing an energy efficient MAC protocol for wireless sensor networks and propose a novel scheme, named as TEEM (Traffic aware, Energy Efficient MAC). Inspired by S-MAC protocol, probably the most well-known sensor MAC scheme for energy efficiency, the proposed TEEM is also based on the concept of 'listen/sleep modes cycle.' However, unlike the S-MAC where the duration of listen and sleep modes are fixed, our TEEM protocol makes these durations adaptive by utilizing 'traffic information' of each node, and hence achieves a significant decrease in power consumption. Through implementation in Motes sensors, the presented TEEM is evaluated and compared to S-MAC as well as the IEEE 802.11-like protocols.**

I. INTRODUCTION

Recently, advancement in MEMS (Micro Electro Mechanical Systems) has enabled the development of smart sensors. A smart sensor has various functions like sensing, processing and networking. A wireless sensor network consists of a certain number of smart sensors with limited battery life and energy expensive short-range radio communication. Due to these energy critical characteristics and high probability of failure, wireless sensor networks need an efficient MAC protocol design. Especially, minimizing power consumption is a primary goal in sensor MAC protocol design because a power drain of each sensor node can cease all the necessary functions of sensor networks.

There has been recent attention on developing energy efficient MAC protocols in wireless sensor networks [1-6]. These protocols can be categorized as either schedule-based [3, 4] or contention-based [1, 2, 5, 6]. Schedule-based protocols are naturally energy preserving in that they have a duty cycle built-in with an inherent collision-free nature, but they often have high complexity in design due to a non-trivial problem of synchronization in wireless sensor networks. Therefore in this paper, we more focus on the contention-based MAC protocol and present a novel scheme, named as TEEM (Traffic aware, Energy Efficient MAC) protocol.

The proposed TEEM is originally inspired by S-MAC [1], probably the most often cited contention-based MAC protocol for sensor networks with the concept of periodic listen and sleep modes – a brief overview of the S-MAC will be provided in the next section. However, unlike the S-MAC where the duration of listen and sleep modes are fixed, our TEEM protocol makes these durations *adaptive* by utilizing 'traffic information' of each node, achieving a significant decrease in power consumption. Thus, with the proposed scheme, the listen time of nodes can be reduced by putting them into sleep state earlier when they expect no data traffic to occur. Note that most of previous works assume to make all nodes be in active for the whole listen period, and only focus on how to enlarging the sleeping time of data packet transmitter and receiver pairs.

For performance evaluation, we have implemented the TEEM over the real Mica Mote sensor nodes platform [9] and compared it with the existing S-MAC and 802.11 protocols. The experimental results demonstrate that our TEEM is superior in terms of energy efficiency.

II. RELATED WORK

One of the famous energy efficient protocols in sensor network is S-MAC [1]. As explained earlier, S-MAC is a contention-based random access protocol with a fixed listen/sleep cycle. It uses a coordinated sleeping mechanism, similar to the power saving mechanism of IEEE 802.11 [7]. A time frame in S-MAC is divided into two parts: one for a listening session and the other for a sleeping session (See Figure 1). Only for a listen period, sensor nodes are able to communicate with other nodes and send some control packets such as SYNC, RTS, and CTS. By a SYNC packet exchange, all neighboring nodes can synchronize together. Then, by the successful RTS/CTS exchange (similar to IEEE 802.11), the two nodes can communicate with each other, meaning that they use their normal sleep time for data packet transmission. All other nodes will simply follow their sleep schedules in order to avoid any wasteful idle listening.

Figure 1 shows the basic scheme of the S-MAC protocol. In this simple figure with three nodes, node A wins the competition for sending out its SYNC packet. Assuming to have data traffic towards node B in its queue, node A again wins the competition for sending a RTS packet (Figure 1(a)). Upon receiving this RTS, node B may acknowledge to node A with its CTS packet.

This work was in part supported by Korea Science & Engineering Foundation, University IT Research Center Project and a grant of the 21C Frontier R&D Program.

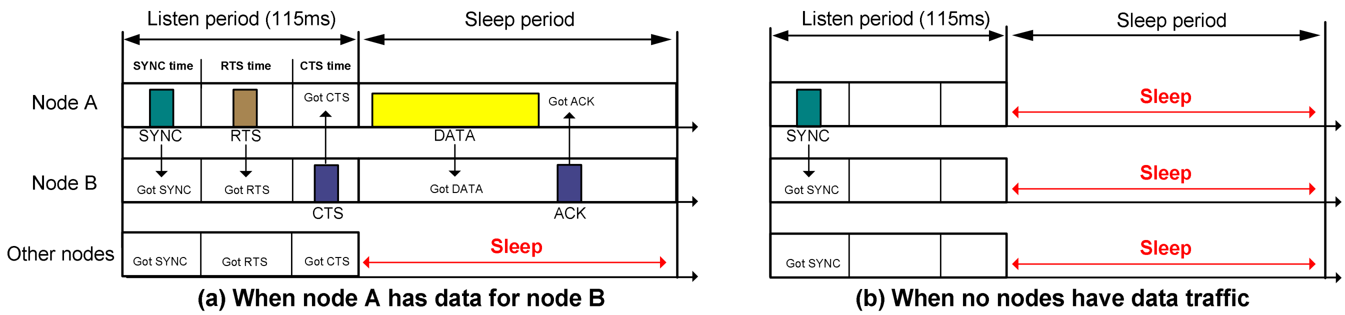

Figure 1. Basic mechanism of S-MAC

Successful exchange of RTS/CTS packets between two nodes implies that they should stay awake in the whole sleep period, followed by the current listen interval, for the completion of their data communication (i.e., until the next listening interval). Again, all other nodes that are not involved in data communication can enter a sleep mode.

Although S-MAC can reduce the idle listening time, it is not optimal due to a *fixed* interval of listening mode. The problem is that, while no nodes have data traffic to send during some time frame and hence no RTS/CTS packet transmissions may occur in the corresponding listen period, every node still has to be awake and just waste their energies – refer to Figure 1(b). Such an inefficiency is caused mainly by the fact that S-MAC (and a descendent of S-MAC like T-MAC[2] and DSMAC[6], as well) does not consider the actual traffic information in the network. This observation leads us to propose a new energy efficient sensor MAC protocol that allows the nodes to go to sleep early even in the listen period when they are aware that nobody has data packets queued at the current time frame.

III. PROPOSED SCHEME

In order to minimize energy consumption, the proposed TEEM makes two important modifications over the existing S-MAC protocol: firstly by having all nodes turn off their radios much earlier when no data packet transfer is expected to occur in the networks, and secondly by eliminating communication of a separate RTS control packet even when data traffic is likely to occur. Remember that in S-MAC, the likelihood of data communication is never considered – thus, every node is required to be in active during the whole listen period. Our goal is to avoid such a fully active listen interval by utilizing traffic information of each node.

S-MAC has the long listen interval, divided into three parts for SYNC, RTS, and CTS packets, respectively. The overall listen interval in the current S-MAC implementation on Mica motes[9] is 115 ms. However, our TEEM has much smaller listen interval of 83 ms, which is divided into only two parts: the first part of the listen interval is for sending SYNC packets when a node has *any* data messages queued for transfer (hence, we call it "$SYNC_{data}$" period), and the second part is also for sending SYNC packets but when a node has *no* data packets to transmit in its buffer (we name it as "$SYNC_{nodata}$" period). That is, in TEEM, each node tries to differentiate its time for broadcasting a SYNC packet. This differentiation is based on whether a node has data traffic or not. Basically, when a node does not have any outgoing data traffic, it will delay a time for sending its own SYNC into the second part of our listen interval, i.e., the $SYNC_{nodata}$ period. It is important to note that there will be no SYNC packets going on at the $SYNC_{data}$ period when no one has outgoing messages currently queued in their buffers. In this case, all nodes will be allowed to sleep right after receiving a SYNC packet at the $SYNC_{nodata}$ period until the next listen schedule, resulting in much longer sleeping time.

The proposed TEEM can further reduce the listen interval of some other nodes that are not involved in data communication, even in the case when there exists node X that has data traffic and thus needs to send its SYNC packet in the first part of the listen interval, i.e., the $SYNC_{data}$ period. Such a reduction of the listening time can be achieved by allowing node X to piggybacking a RTS onto its SYNC packet. It is possible to combine these two control packets in our protocol since, if node X is assumed to win the channel contention for sending its SYNC packet in the $SYNC_{data}$ period, node X should also be assumed to send out its data messages. This combined control packet between SYNC and RTS is called a $SYNC_{rts}$ packet. Figure 2 represents a $SYNC_{rts}$ packet structure. It looks quite similar to a SYNC packet, just borrowing two additional fields for the address of the receiver and NAV duration from a RTS. In our test-beds using Mica Mote, only 4 bytes are additionally required for the $SYNC_{rts}$ packet.

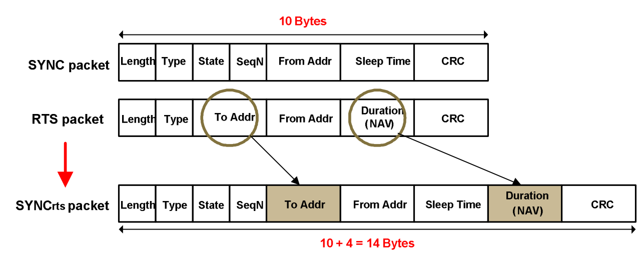

Figure 2. SYNCrts packet type: A combination of SYNC and RTS

When nodes receive $SYNC_{rts}$ packet by the end of $SYNC_{data}$ time interval, they will come to know not only synchronization information, but also information about who is the receiver for data packets. Therefore, while the destined receiver is required to reply a CTS packet back to the sender, any other nodes are free to go to sleep without necessarily staying awake in the $SYNC_{nodata}$ period – thus, they are needed to be in active only for the first, very short duration of listen interval. In this sense, the duration of listen and sleep modes in our TEEM scheme is not fixed but adaptive.

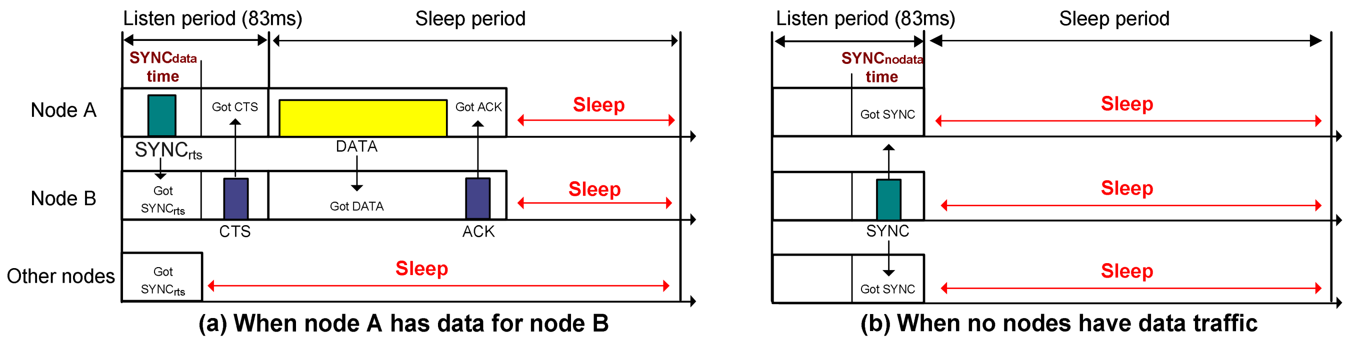

Figure 3. Basic mechanism of the proposed TEEM

See Figure 3 as an example. In Figure 3(a) where node A has data messages to send for node B, it broadcasts the $SYNC_{rts}$ control packet during the first part of listen period. Assuming this control packet is successfully transmitted, the receiver node B will detect it and prepare to reply its CTS packet, whereas other nodes become aware of the fact that they are not destined for and hence go to sleep until the next listen schedule. Of course, nodes A and B will use their normal sleep time for data packet transmission without following their sleep schedule. Observe that they will start to sleep after they complete the transmission. In wireless sensor networks with light traffic load, there are often cases when no sensing events occur in the network. Figure 3(b) represents these cases. As explained earlier, every node delays a time for broadcasting a SYNC packet into the $SYNC_{nodata}$ period. When no such SYNC packet exists in the first part of listen period, nodes are all aware of the fact that no data traffic will be at this time and therefore simply follow their sleep schedules.

IV. Performance Evaluation

The purpose of the experiments is to measure the energy consumption of three contention-based MAC protocols: IEEE 802.11, S-MAC and our TEEM. Our tested sensor node is "Mica Mote" which has been developed at U.C. Berkeley and now commercially available from Crossbow [9]. It is equipped with a low-power microprocessor (Atmel 90LS8535), 128K of program memory, 4 K of SRAM, and narrowband RF device for wireless communication. Direct communication with a host computer is possible through a serial port. The performance of S-MAC was analyzed on Mica Mote in [1]. Which is currently one of the standard communication stacks in TinyOS[8] operating systems. For the purpose of performance comparison, we implemented our TEEM modules on Mica Motes. To measure the energy consumption, we define three different energy modes: receiving, transmitting and sleep. We set the energy model based on S-MAC which consumes 13.5mW, 24.75mW and 15uW, per receiving, transmitting and sleeping modes, respectively. Again, we compared three MAC modules, as listed below.

1. 802.11-like protocol without sleep
2. S-MAC protocol
3. TEEM protocol

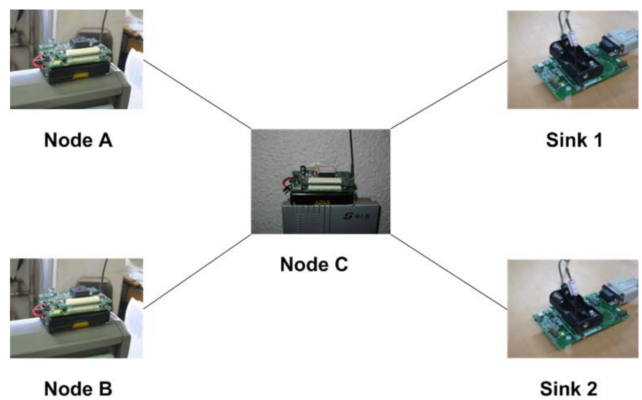

Figure 4. Two-hop Network Topology with Mica Motes

The first model above consists of several pieces as in IEEE 802.11 DCF: random back-off and retrial, RTS/CTS/DATA/ACK packet exchange, and fragmentation support. The duty cycle of both S-MAC and TEEM is selected as 50%, just like in [1].

To evaluate the performance of three protocols, we perform the tests on a simple topology of two-hop networks with two sources and two destinations as illustrated in Figure 4. Data flows pass through from node A to Sink 2, and from node B to Sink 1 both via the intermediate node C. We change the inter-arrival period of messages in order to measure energy consumption in different traffic load. In our experiment, the message inter-arrival period varies from 1 to 7 seconds, meaning that a message is generated every 1 (to 7) seconds by each source node. 20 messages with 100 bytes each are periodically generated to be transferred to each sink node. For example, if the message inter-arrival period is 7 seconds in one test, that test will finish after 140~150 seconds because source node will generate 20 messages every 7 seconds and we stop test when both sources send 20 messages to each sink node. For performance evaluation, we measure three metrics such as the percentage of sleep time of the source nodes, a total number of control packets (i.e., RTS, CTS, and ACK) measured at source, and total energy consumption of source node when using different MAC protocols.

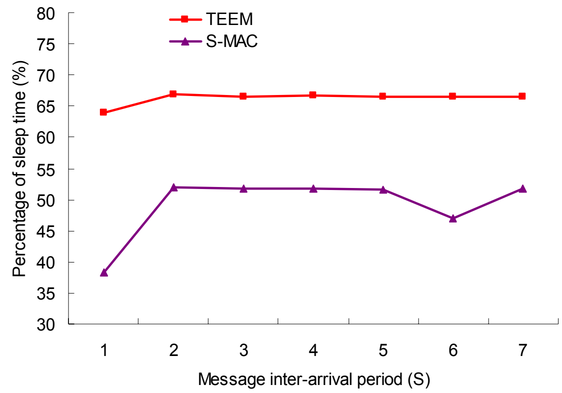

Figure 5. Percentage of time in the sleep mode

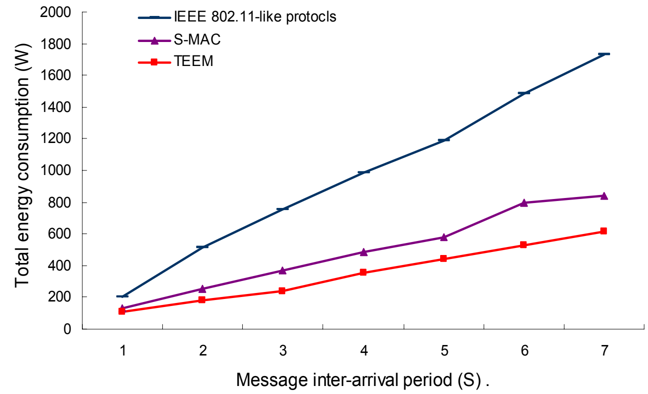

Figure 7. Energy consumption in soure node

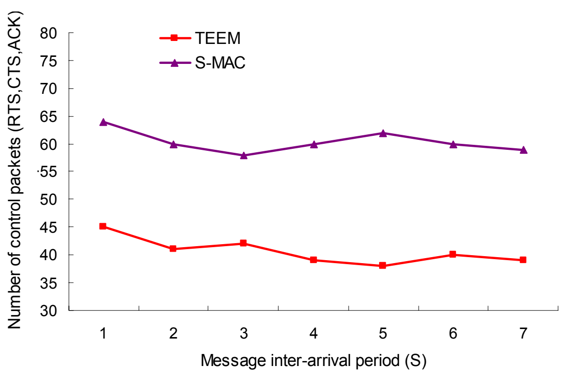

Figure 6. Number of control packets

Figure 5 shows that the percentage of time that the source nodes are in the sleep mode. In this experiment, we compare only two MAC protocols having sleep modes (i.e., S-MAC and TEEM). Recall that IEEE 802.11 has no sleep mode. Because our TEEM has more adaptive listen period, its sleep time is constantly longer than S-MAC as shown Figure 5. It indicates that our scheme is likely consuming less energy than S-MAC.

Figure 6 shows the number of control packets counted at a source in S-MAC and TEEM. We measure the total number of RTS, CTS and ACK packets transmitted as well as received at the source node. In our scheme, when there is data traffic, RTS packet is piggybacked into SYNC packet ($SYNC_{rts}$), so no RTS packets are sent. Therefore, TEEM results in much smaller number of control packets than S-MAC.

Figure 7 also shows the energy consumption from a source node. According to the figure, when the message inter-arrival period is increased, TEEM uses less energy consumption than either S-MAC or 802.11. We believe, this is because our scheme has much shorter listen period than the others and hence can sleep earlier than S-MAC. Overall, our TEEM can save energy significantly and works more efficiently. For instance, if the message inter-arrival period is 6 seconds in Figure 7, TEEM saves as much as 75% and 35% of the energy over IEEE 802.11 and S-MAC.

V. Conclusion

In this paper, we propose a new energy efficient MAC protocol for wireless sensor networks, named the TEEM (Traffic aware, Energy Efficient MAC). To reduce energy consumption, TEEM is also based on the concept of 'listen/sleep modes cycle' like S-MAC. However, TEEM is more energy efficient because it has much shorter and adaptive listen period by utilizing 'traffic information'. We have implemented the TEEM on the Mica Motes hardware, and evaluated it with the real test-bed sensor nodes. Our experimental results demonstrate that our scheme works well and saves significant energy compared to S-MAC or 802.11 schemes.

References

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

[2] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," *in ACM Sensys'03*, Nov. 2003.

[3] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J. Pottie., "Protocols for self-organization of a wireless sensor network," *in IEEE Personal Communications*, pp.16-27, Oct. 2000.

[4] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *in ACM Sensys'03*, Nov. 2003.

[5] W. Ye, J. Heidemann, and D. Estrin "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," *in IEEE Transactions on Networking*, June 2003.

[6] P. Lin, C. Qiao, and X. Wang "Medium Access Control With A Dynamic Duty Cycle For Sensor Networks," *in WCNC*, Mar 2004.

[7] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, 1997.

[8] TinyOS, http://webs.cs.berkeley.edu/tos/

[9] Mica Mote, http://www.xbow.com/