AN ENERGY EFFICIENT TDMA MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

Priyanka Kaushik

# M-Tech Scholar, Department of electronic and communication engineering, Punjab University, NITTTR, 26 sec, Chandigarh
priyankakaushik1985@gmail.com

Abstract: In this paper an energy efficient TDMA MAC protocol for wireless sensor networks is presented. Wireless sensor networks use battery operated computing and sensing devices. A network of these devices will collaborate for a common application such as environmental monitoring. This protocol uses TDMA to give nodes in the WSN the opportunity to communicate collision free, the network is self-organizing in terms of time slot assignment and synchronization. In addition, the multi-hop mechanism has been developed between cluster head nodes and base station to ensure long distance transmission.

Keywords- Medium Access Control (Mac), Wireless Sensor Network (Wsn), Time Division Multiple Access (Tdma), Latency, Throughput.

I. INTRODUCTION

Improvement in hardware technology has resulted in low-cost sensor nodes which are composed of a single chip with embedded memory, processor, and transceiver. Low power capacities lead to limited coverage and communication range for sensor nodes compared to other mobile devices. Hence, for example in target tracking and border surveillance applications, sensor networks must include a large number of nodes, to cover the target area [1].

Wireless sensor networks (WSNs) have widely been used in many fields such as environmental monitoring, intelligent building management and product quality monitoring, etc. They can provide an easier interaction between man, machine and environment, which make our life easier [2]. In most WSN applications, however, nodes are usually powered by batteries and the nodes will become inactive, even more to losing their sensing and communication functionalities when energy is depleted. The transmission distance in WSNs is about 100m~1000m and it will be shortened by the reinforced concrete wall. This problem can be solved by multi-hop way. Therefore, it is of great importance to increase the reach of communication range and it is also of great importance to reduce the energy consumption of WSNs and prolong the network lifetime in the practical applications [3].

II. RELATED WORK

In WSNs, the medium access layer controls the radio, and it has a large impact on the overall energy consumption. Although the research field of WSNs is relatively new, some interesting studies to MAC protocols for this type of networks can be found in literature. In this section we will describe some of these protocols e.g. SMAC, TMAC, LMAC and EMAC for comparison [4].

The sensor-Mac (SMAC) protocol [4] recognizes two phases in transceiver usage of network nodes: a listen period and a sleep period. In the sleep period, the nodes turn off their power consuming transceiver. After the sleep period, the nodes wake-up and listen whether communication is addressed to them, or they initiate communication themselves. This implies that the sleep and listen periods should be (locally) synchronized between nodes. Because the protocol is carrier sense multiple access with collision detection (CSMA/cd) based in the listen period, synchronization does not have to be very strict and nodes can use their sleep period as well for communication if needed. To prevent collisions of short “SYNC” messages (used for synchronization), which only contain a identification number of the sender and the next time nodes goes to sleep, the SMAC protocol divides the listen period in two sections. The first part is reserved for SYNC messages and the other part is reserved for request to send (RTS) messages. The SMAC protocol is also capable of transmitting Omni cast messages [4].

Timeout T-MAC [3] is the protocol based on the S-MAC protocol in which the Active period is pre-empted and the sensor goes to the sleep period if no activation event has occurred for a time ‘Ta’ as shown in Fig. 2. The event can be reception of data, start of listen/sleep frame time etc. The time ‘Ta’ is the minimal amount of idle listening per frame. The interval Ta > Tci + Trt + Tta + Tct where Tci is the length of the contention interval, Trt is the length of an RTS packet, Tta is the turn-around time (time between the end of the RTS packet and the beginning of the CTS packet) and Tct is the length of the CTS packet. The energy consumption in the Timeout TMAC protocol is less than the Sensor S-MAC protocol. But the Timeout T-MAC protocol has high latency as compared to the S-MAC protocol [6].

LMAC is a TDMA-based protocol for WSNs that give nodes the opportunity to communicate collision-free [8].TDMA protocols are centralized or distributed scheduled TDMA systems provide a natural way to conserve energy [6]. TDMA-based protocol divides time into time slots. Each slot can be assigned by only one node and this node controls over this time slot, so each node can use its own time slot to transfer data without having to content for the medium or deal with energy wasting collision of transmissions. However, it requires accurate time synchronization between the access point and the individual nodes to ensure that a node can be waked up exactly at the start of “its” slots. Therefore, TDMA based protocols need a good synchronic scheme. Such schemes are not easy to be implemented in dynamic WSNs.
The TDMA-based EMACs protocol divides time into *time slots*, which nodes can use to transfer data without having to contend for the medium or having to deal with energy wasting collisions of transmissions. A node can assign only one slot to itself and is said to control this slot. After the frame length, which consists of several time slots, the node again has a period of time reserved for it. A time slot is further divided in three sections: Communication Request (CR), Traffic Control (TC) and the data section. In the CR section other nodes can do requests to the node that is controlling the current time slot. Nodes that have a request will pick a random start time in the short CR section to make their request. These messages are comparable to RTS messages in SMAC. Communication in this section is not guaranteed collision-free. Nodes that do not have a request for the current slot owner, will keep their transceiver in a low power state during the entire CR section. The controller of a time slot will always transmit a TC message in the time slot. When a time slot is not controlled by any node, all nodes will remain in sleep state during that time slot [4].

**III. PURPOSED WORK**

The concentration of this work is on the protocols’ ability to transfer data as efficiently as possible; the communication protocol in each cluster relies on the cluster head. It is developed to be pre-scheduled and every node in the cluster only transmits in its time slot and sleeps in other time-slot. Thus, a new node can be added into networks by schedule algorithm to cope with the network topology changes. Finally, the cluster head transmit data through multi hop to base station.

Sensors equipped with transceiver, processor and memory will be deployed by the millions. Hence the costs of a single smart sensor must be at a minimum. This does not only translate to scarce resources –like energy and memory- in the sensors, but also to complexity of the hardware. Currently, multi-channel transceivers are available on the market, but they are still higher priced than single channel versions.

During the design of the medium access protocol, we assumed a single channel transceiver, which has three operational states: transmit, receive and standby. Typically, transmitting consumes more power than receiving and standby lies beneath the power consumption of receiving by a factor 1,000 or more. Summarizes some parameters of a transceiver we use for prototyping. These parameters are also used in our physical layer model in the simulator to obtain network lifetime results.

**TABLE 1. TRANSCEIVER RATE (RFM TR 1001)**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy consumption Tx</td>
<td>21mW</td>
</tr>
<tr>
<td>Energy consumption Rx</td>
<td>14.4mW</td>
</tr>
<tr>
<td>Energy consumption sleep</td>
<td>15mW</td>
</tr>
<tr>
<td>Energy consumption sleep/Tx</td>
<td>16 μW</td>
</tr>
<tr>
<td>Energy consumption sleep/Rx</td>
<td>518 μW</td>
</tr>
</tbody>
</table>

LMAC divides a time frame into 32 equal time slots. Because a time slot can only be controlled by a single node, this node can communicate collision-free. Each time slot consists of a control section and a data section. The control section has a fixed size (12 byte) and is used for several purposes [5]. It carries the ID of the time slot. It indicates the distance of the node to the base station in hops and addresses the intended receiver and reports the length of the data section. The transmission of the control section is carefully timed by the nodes. In the data section, each sensor node can transmit information in their time slot and works effectively.

In our protocol, time is divided into cyclical time frame and each frame is divided in three parts: frame synchronization slot, work slots and request slot as shown in Fig. 2. The function of frame synchronization slot insures the sensor nodes synchronize with the cluster head. The cluster head broadcast synchronic beacon at the beginning of each frame. The synchronic mechanism will be discussed in detail. Each sensor node in the network occupies a work slot and transmits information when the node is in its turn. In LMAC, there are 32 work slots for sensor nodes. It is unsuited for adding new unknown sensor nodes into the network. We add a request section after data section in each work slot. A new sensor node sends RTS message toward master node and gets ACK message to add the network in next frame. The network topology can be dynamically changed by this way.

**TABLE 2. CONTENTS OF THE CONTROL SECTION**

<table>
<thead>
<tr>
<th>Description</th>
<th>Size(Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>2</td>
</tr>
<tr>
<td>Current Slot number</td>
<td>1</td>
</tr>
<tr>
<td>Destination ID</td>
<td>2</td>
</tr>
<tr>
<td>Data size(Bytes)</td>
<td>1</td>
</tr>
<tr>
<td>Distance to Gateway</td>
<td>1</td>
</tr>
<tr>
<td>Collision in slot</td>
<td>1</td>
</tr>
<tr>
<td>Occupied Slot</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>12</strong></td>
</tr>
</tbody>
</table>
A. Cluster Time Slot Assignment
Slot assignment can be done by using different strategies. One of these strategies is predetermined slot assignment. This is the easiest way and suitable inside the cluster. Each sensor node in the cluster is assigned a fixed ID. As shown in Fig. 2, frame synchronization slot locates the start of each time frame. In frame synchronization slot, the cluster head sends timer synchronization (SYNC message) and network commands. The work slots consist of 32 time slots for fixed sensor nodes and another 8 time slots for new added sensor nodes. The cluster head listens to the channel and waits for the RTS message of a new added sensor node in the request slot.

There are 12 bytes in the control section according to LMAC. In our protocol, the length of a control section is reduced to 6 bytes as shown in Table I. Identification will also be used to maintain slot synchronization between the sensor nodes and the cluster head.

<table>
<thead>
<tr>
<th>Description</th>
<th>Size(Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>2</td>
</tr>
<tr>
<td>Current Slot number</td>
<td>1</td>
</tr>
<tr>
<td>Destination ID</td>
<td>2</td>
</tr>
<tr>
<td>Data size(Bytes)</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>6</strong></td>
</tr>
</tbody>
</table>

The number of TDMA cycles (NR) is calculated as in

\[ N_r \leq \frac{1}{2} \left( \frac{T_g}{T_{frame} \theta} \right) \]

Where \( T_g \) is the slot guard time, \( T_{frame} \) is the frame time and \( \theta \) is the sensor clock accuracy.

B. Time Synchronization in Cluster
The time synchronization is the key technology for TDMA based MAC protocols. Over the years, many protocols have been designed for maintaining synchronization of physical clocks. The aim of the time synchronization protocol is to improve the precision of time synchronization and to upgrade the efficiency on energy for WSNs. There are several synchronization algorithms, such as RBS [11], TPSN, DMTS [10], and so on.

In DMTS a leader is selected as time master and broadcasts its time. All the receiver devices measure the time delay and set their time as received master time plus measured time transfer delay. As a result, all the devices that have received the time synchronization message can be synchronized with the leader. The time synchronization accuracy is bounded mainly on how well the delay measurements are along the path.

DMTS can be extended to multi-hop sensor network. If a node knows that it has child/children, it broadcasts a time signal after it adjusts its own time. This situation becomes a single hop time synchronization problem with a second level leader. However, in some WSNs, a node has no knowledge of its child/children. To tackle this issue, we propose the following multihop time synchronization algorithm.

**Sender:**

Send TimeSync, MAC preamble

Send Start symbol

Send data

Receive ACK

**Receiver:**

Receive preamble

Receive data

Send ACK

Rx process

The sender’s and receiver’s radio device will synchronize with each other in the end of the start symbol. If a receiver take a local timestamp when its radio synchronized with the sender’s radio, and take another local timestamp when processing the time message, it then has measured both the data transfer time (excluding preamble and start symbols) and receiver side processing delay. Figure 1 shows the time line of transfer a time message from one node to another in mica hardware platform. When the radio propagation delay is neglected, the total delay \( t_d \) can be measured as:

\[ t_d = t_e + (t_2 - t_1) \]

Where \( t_e \) is the estimated time to transmit the preamble and start symbols, \( t_2 \) and \( t_1 \) are receiver timestamps. Since a radio device has a fixed transmit rate, for example, Mica radio transmit preamble and start symbols at the speed of 20 kbps, \( t_e \) is a fixed delay and can be calculated as

\[ t_e = n \tau \]

Where \( n \) is the number of bits to transmit and \( \tau \) is the time to transmit one bit over radio. In DMTS, a time synchronization leader sends a time synchronization message with its timestamp \( t \), which is added after MAC delay and a clear channel is detected. The receiver measures the path delay and set its local clock to \( t_l \).

\[ t_l = t + n \tau + (t_2 - t_1) \]

The receiver is then synchronized with the leader. The lower bound of DMTS is the radio device synchronization accuracy, and the upper bound is the precision of local clock.

Since only one time signal transfer is required in DMTS to synchronize all nodes within a single hop. This method is energy efficient. It is also lightweight because there are no complex operations involved. From the receivers’ points of view, the sender’s time signal is a common-view timestamp, hence all the receivers synchronize with each other better than with the sender. This is mainly because the transmit time estimation error is cancelled out.
C. Multi-hop mechanism between cluster heads and base station
We use multi-hop communication between cluster heads and base station to ensure long distance information transmission.

- Link table establishment. The same link table is developed in the nodes with the same domain name. As a result, when node needs to be changed, the setup of link address and link table of the new node can be copied from the original one.

- The base station is defined as the first order node. The second order node, and so forth. The first order node as the primary node can actively initiate communication. The second order node works as a primary node only when the first order node is disabled. The multi-hop relay wireless communication link structure is shown in Fig. 3.

![Multi Hop Relay Wireless Communication](image)

- According to the multi-hop setup of WSNs mentioned above, the waken instruction is actively and timely sent by the primary node to wake up the neighboring and indirect layer nodes and the data is sent to the neighboring layer nodes. Then the neighboring layer nodes work as main node to send the data of the primary node and that of themselves and so forth. The data transmissions are completed when the last order node is woke up and the data from every layer node are saved in the link table of the last order node.

- If the communication between the first order node and the second order node is failed, a new communication will be initiated after one transmission time. If two such communications are failed, the second order node will be considered as damaged. Then the third order node will be tried by the first order node, which is also limited within twice.

- When non-first order node receives data from higher order node, the data will be stored in the link table. An acknowledgement signal will be sent out and the data will continue to be sent to the lower order node. When the data from indirect layer node is received, whether a successful transmission occurred between the neighboring layer nodes is determined first. If transmission is successful, the data from indirect layer node will not be processed; otherwise the data from indirect layer node are received, stored in link table and return an acknowledgement signal. After that the neighboring layer higher order node will be accessed, the data from which will be stored and sent down to lower order nodes.

- If the transmission from the first order node fails, e.g., the acknowledgement signal from the second order node is not received by the first order node, the second order node will be woken again by the first order node after a transmission time of neighborhood layer node. If the acknowledgement signal from the second order node is again not received, the first order node will try the third order node for indirect layer node communication. Once the transmission to the third order node fails, the whole communication will be interrupted and alarm information is sent out.

IV. CONCLUSIONS
In this paper we have described a TDMA-based MAC protocol in the cluster and the multi-hop mechanism between cluster heads and base station. In the cluster, the operation is dependent on the cluster head. We design our protocol by reference to LMAC. DMTS time synchronization algorithm is used. The sensor nodes have own time slot number, they can communicate with the master node without collision problem. The new sensor node can be added into the network easily. The multi-hop communication between cluster heads and base station is designed to ensure long distance information transmission. The future work will be focused on research of secure protocols for the communication.

REFERENCES


