ITEM - Implementation of Integrated TDMA and E-ASAP Module

Inderjit Singh
CTU Faculty of Electrical Engineering
Department of Control Engineering
Karlovo nám. 13, Prague 2, Czech Republic
inderjit.arora@gmail.com

Jiří Trdlička
CTU Faculty of Electrical Engineering
Department of Control Engineering
Karlovo nám. 13, Prague 2, Czech Republic
trdlij1@fel.cvut.cz

Zdeněk Hanzálek
CTU Faculty of Electrical Engineering
Department of Control Engineering
Karlovo nám. 13, Prague 2, Czech Republic
hanzalek@fel.cvut.cz

Abstract

We present a new implementation of TDMA communication protocol for wireless sensor networks. The ITEM (Integrated TDMA E-ASAP Module) is a TinyOS 1.x module primarily determined for TelosB platform. It provides a TDMA (Time Division Multiple Access) communication protocol integrated with slot assignment protocol E-ASAP (Extended - Adaptive Slot Assignment Protocol). The ITEM module deals with a collision free multi-hop communication, hidden-node problem and network structure changes. To achieve a good data throughput the E-ASAP sets the TDMA period individually for each node depending on the actual network structure. The current version of the ITEM module is implemented for TelosB and TmoteSky platforms.

1. Introduction

The Wireless Sensor Networks (WSN) is a very quickly developing technology. There are several different platforms and operating systems. One of the most popular operating systems today is TinyOS [1] for which many applications and modules have been implemented. However, according to our knowledge, there is no TDMA implementation in TinyOS suitable for the TelosB platform [2].

Decision whether the TDMA communication is better than the CSMA depends on the concrete application. However, the TDMA eliminates the data collision during the communication and thereby it decreases the communication energy consumption and holds a constant data throughput independent of the medium load. Simultaneously, the collision-free communication ensures a constant one-hop communication delay between two nodes, which is very important for the real-time communication. The real-time constraint was our main motivation to implement the TDMA module. Due to the knowledge of the timing of the TDMA, the nodes of the WSN can switch to a sleep mode and save much more energy.

An important part of the TDMA mechanism is a slot assignment protocol, which assigns the slots to the nodes, so that no node can interfere with a communication of different nodes. For this task, we have adopted the E-ASAP (Extended - Adaptive Slot Assignment Protocol) which has been published in [6]. The E-ASAP is an on-line distributed protocol, which takes care about the hidden node problem and individually sets the TDMA period for each node to achieve a better data throughput. The protocol is able to dynamically adapt the slot assignment according to the actual network structure. Especially the dynamic behavior of the E-ASAP is very important, because the structure of the WSN is changing in many applications.

There is a lot of works focused on the TDMA protocols for the sensor networks (e.g. see [9]). However, we were able to find just some few implementations with a source code for the TinyOS: Lin Gu has implemented a TDMA protocol PRIME in TinyOS 1.x for Mica2 platform. The implementation of the PRIME protocol can be found in contrib/prime of TinyOS 1.x [3]. In [7] the authors propose a Unified Power Management Architecture (UPMA) for flexibly integrating the use of different radio power management protocols into a complete wireless sensor network system. The UPMA has been added into the contrib/wustl/upma of TinyOS 2.x [4]. We should mention the open implementation of the IEEE 802.15.4 with the GTS allocation too [8].

The paper is organized as follows: Section 2 introduces the basic ideas of the E-ASAP protocol. In Section 3 the structure and functions of the ITEM is presented. The example of ITEM behavior is given in Section 4 and the Section 5 concludes the work.

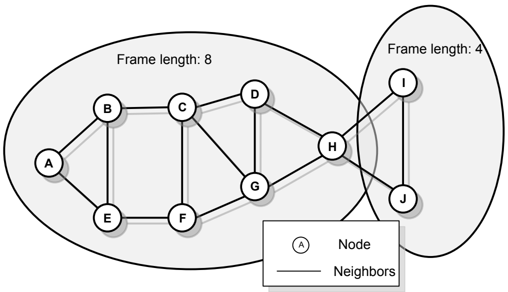

Figure 1: Example of a WSN.

Own

Slot(s) / Frame length
2 / 8

Neighbors

Node	Slot(s) / Frame length
D	6 / 8
G	4 / 8
I	1 / 4
J	3 / 4

Hidden nodes

Node	Slot(s) / Frame length
C	1 / 8
F	5 / 8

Figure 2: Information held in node **H** in Figure 1.

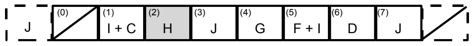

Figure 3: TDMA schedule in node **H** view in Figure 1.

2. E-ASAP

In this section, we briefly introduce the E-ASAP (Extended - Adaptive Slot Assignment Protocol). For more detailed description see [6].

The TDMA protocol divides the usage of the communication channel into several non-overlapping slots, which are repeated within some period. The number of slots in one period is called "**frame length**". The E-ASAP sets the frame length of a new node (node joining the network) and assigns a free slot to this node. Eventually, if there is no free slot, the E-ASAP doubles the frame length of the interested nodes.

2.1. Frame Format

The frame length is set as a power of two in E-ASAP and the minimum frame length is fixed to four slots. By setting the frame length as a power of two, packet collision can be avoided between nodes with different frame lengths. The first slot (slot zero) is reserved for a new node to transmit a request for doubling the frame length of its neighbor nodes if its needed. An example of a TDMA frame is on Figure 3.

2.2. Data Format

Each node maintains the information about the frame length and slots assignment of itself, its neighbors and its hidden nodes (nodes, which cannot communicate each other and have a common neighbor). An example of information held by the node **H** in Figure 1 is shown on Figure 2. The slot means the number of the assigned slot and the frame length means the number of slots in the TDMA period of the node. An example of the TDMA schedule in view of node **H** is shown on Figure 3. The slot zero is free and reserved for a new node. The letters in the slots denote the nodes, which can send data in a given slot. There are assigned two nodes in slot number one. It is possible, because the transmission from node **I** and from node **C** do not interfere with each other.

2.3. Packet Format

There are two types of data packets in the E-ASAP. A data packet (DAT) and an information packet (INF). The DAT packet contains the information on the frame length of the sender, the current slot number, the maximum frame length among the sender and its neighbors and the transmitted data. The INF packet contains the information about the network structure in the node neighborhood. There is information about the frame length and the assigned slots of the sender and its neighbors. e.g. in Figure 1, the INF packet of node **H** contains the fist and the second table from Figure 2 (Own, Neighbors).

2.4. Slot Assignment

The newly joined node collects INFs transmitted by its neighbors to obtain the slot assignment information. After some period , the new node sets its frame length to four slots (minimum frame length) and finds a free slot via the following procedures:

1. *Getting an unassigned slot*
   If the first slot is not assigned to any neighbor, and there is an unassigned slot in the TDMA period (except the first one), the new node assigns one of the free slots to itself (except the first one).

2. *Doubling the frame length*
   If no slot is available in the current frame length, the new node doubles the frame and tries again to assign an unassigned slot. This procedure is repeated until the new node finds an unassigned slot, or until the maximum allowed frame length is reached.

After selecting a slot, the new node sends the INF packet to its neighbors. The INF packet contains the information about itself and about the node neighbors.

2.5. Releasing Slot Assignment

When a node exits from the network, it just stops sending the DAT and the INF packets. The neighbors detect departure of the node after some time when no DAT or INF packet has been received. Each neighbor changes its assignment information and sends them by an INF packet. If possible, each node decreases its frame length.

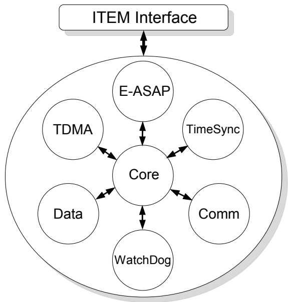

Figure 4: ITEM structure

3. Implementation

The ITEM is designed to be flexible and easy to modify. In this section, we present the structure of the ITEM and functions of the modules. The full documentation with functionality diagrams, user interfaces and TinyOS documentation can be found in [5].

3.1. Structure

The structure of the ITEM is divided into seven modules (see Figure 4) and each module is implemented as an independent component. This does not hold for the Core module, which interconnects all other modules to make them cooperate. This structure enables an easy modification and improvement of the ITEM.

A brief description of the modules follows:

E-ASAP The E-ASAP implements the E-ASAP protocol [6]. It handles the network information about the node, its neighbors and its hidden nodes and updates this information according the network changes. The E-ASAP module takes care about the slots assignment, slots releasing and frame length changes (see section 2).

TDMA The Time Division Multiple Access module manages the TDMA period in terms of timing. It handles the information about duration of the slots and about the TDMA period. It signalizes the beginning of a new slot and enables changes of the TDMA parameters.

Comm Comm module is responsible for all communications related to other nodes via radio and host computer via the USB. It has a basic functionality. Either it can be requested to transmit a message, or when a messaged is received, an event is signaled to all other modules.

TimeSync The TimeSync module handles the time synchronization between the nodes. For the time synchronization in the network, the module uses a simple algorithm:

$$t_{Local} = \frac{t_{remote} + t_{local}}{2} \quad (1)$$

Where t_{Local} denotes the new node local time, t_{remote} denotes the time received from a neighbor node and t_{local} denotes the old node local time. The actual local node time is send with each DAT and INF package, so the time update is performed during each message reception. The same algorithm has been used e.g. in [10] for synchronization of a hexagonal sensor network. The worst-case error (time difference between two neighbor nodes) of this algorithm was 10 ms in our experiments (TelosB platform, TinyOS 1.x). The error is cased by the fact, that the algorithm does not take into account the message propagation delay. We intend to improve the time synchronization algorithm in future work. However, we consider if we are able to achieve better accuracy because of the non-Realtime behavior of the TinyOS.

Data Data module has two queues that hold the data for both reception and transmission. If any of the queues is full, new data are discarded until an empty queue element is available. The module provides functions to manipulate with the data in the queues.

Watchdog As any watchdog, its main function is to reset the system if it locks itself up due to an error. Please note that the watchdog only works for the MSP430 architecture in this implementation.

Core Core module's function is to combine all the other modules in ITEM and make them work together appropriately.

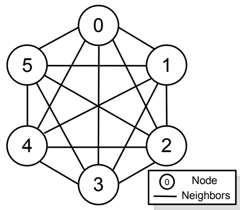

Figure 5: Network structure for a test of ITEM dynamic. (without hidden nodes)

4. Experiments

We present two experiments focused on the slots assignment and the frame length behavior. The nodes are added into the network and removed from it and the slots assignment and the frame length are monitored. We present an experiment in a network, where each node can communicate with all the other nodes (no hidden nodes) at first. Than we present the same experiment in a network with hidden nodes.

4.1. Network without Hidden Nodes

The network structure for the firs experiment is shown on Figure 5. The experiment has been initiated by node 0. The table with slot assignment and frame length progress is in Table 1. You can see the changes of the frame length in the table. All nodes in the network have the same frame

Table 1: ITEM dynamic in network without hidden nodes.

	Node ID (slot / frame length)					
Action	**10**	**11**	**12**	**13**	**14**	**15**
Started 0	1/4	-	-	-	-	-
Added 1	1/4	2/4	-	-	-	-
Added 2	1/4	2/4	3/4	-	-	-
Removed 1	1/4	-	3/4	-	-	-
Added 3	1/4	-	3/4	2/4	-	-
Added 1	1/8	4/8	3/8	2/8	-	-
Added 5	1/8	4/8	3/8	2/8	-	5/8
Removed 3	1/8	4/8	3/8	-	-	5/8
Added 4	1/8	4/8	3/8	-	2/8	5/8
Added 3	1/8	4/8	3/8	6/8	2/8	5/8
Removed 1	1/8	-	3/8	6/8	2/8	5/8
Removed 5	1/8	-	3/8	6/8	2/8	-
Removed 3	1/4	-	3/4	-	2/4	-
Removed 4	1/4	-	3/4	-	-	-

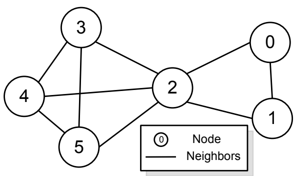

Figure 6: Network structure for test of ITEM dynamic. (with hidden nodes)

length. It is caused by the fact that all nodes can communicate with each other and interfere with each other.

4.2. Network with Hidden Nodes

The network structure for the second experiment is shown on Figure 6. The node 2 works as a gate between two parts of the network. These two parts cannot directly communicate, however the nodes of both parts are neighbors of the node 2. The experiment has been initiated by the node 0 and the dynamics of the network is shown in Table 2.

There is an example of different frame lengths in the network in the 4th row of the table. The node number 4 did not need to increase its frame length until the node number 3 was added. The expression $3(7)/4$ means that the node has assigned the slot number 3. However, for nodes with frame length equal to 8 it seems that the node 4 has assigned the slots 3 and 7.

5. Conclusion

We have introduced a new implementation of a TDMA communication protocol for TinyOS 1.x and TelosB platform, called ITEM. The module uses the E-ASAP to assign TDMA slots to nodes and provides an autonomous control of the slot assignment in the network. The protocol adapts the slot assignment according the changes in the network structure (appear / disappear of the nodes).

Table 2: ITEM dynamic in network with hidden nodes.

	Node ID (slot / frame length)					
Action	**10**	**11**	**12**	**13**	**14**	**15**
Started 0	1/4	-	-	-	-	-
Added 2	1/4	-	2/4	-	-	-
Added 4	1/4	2/4	-	3/4	-	3/4
Added 1	1/8	4/8	2/8	-	-	3(7)/4
Added 3	1/8	4/8	2/8	5/8	3/8	-
Added 5	1/8	/8	2/8	5/8	3/8	6/8
Removed 4	1/8	4/8	2/8	5/8	-	6/8
Removed 1	1/8	-	2/8	5/8	-	6/8
Removed 3	1/4	-	2/8	-	-	6/8
Removed 5	1/4	-	2/4	-	-	-

Moreover, the protocol adapts the frame length (TDMA period) to obtain a more efficient data throughput.

We are working on a new version of the ITEM now. The new version will be implemented in TinyOS 2.x and it will improve interface and some functions to be more user friendly. We intend to use the ITEM module in our next applications from sensor networks area, especially to ensure a real-time behavior of the network and to set the nodes into the sleep mode.

Acknowledgement: This work was supported by the European Commission under project FRESCOR IST 034026.

References

[1] [Online]:http://www.tinyos.net/.
[2] [Online]:http://www.xbow.com/Products/ /productdetails.aspx?sid=252.
[3] [Online]:http://tinyos.cvs.sourceforge.net/tinyos/tinyos-1.x/contrib/prime.
[4] [Online]:http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/wustl/upma/.
[5] [Online]:http://rtime.felk.cvut.cz/∼trdlij1/doku.php?id=ITEM.
[6] A. Kanzaki, T. Hara, and S. Nisho. An adaptive tdma slot assignment protocol in ad hoc sensor networks. *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1160 – 1165, 2005.
[7] K. Klues, G. Xing, and C. Lu. Towards a unified radio power management architecture for wireless sensor networks. *in WWSNA*, 2007.
[8] A. Koubaa, M. Alves, and E. Tovar. GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks. *The 14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'06), Rhodes, Greece*, April 2006.
[9] A. Rowe, R. Mangharam, and R. Rajkumar. Rt-link: A time-synchronized link protocol for energy constrained multi-hop wireless networks. *Third IEEE International Conference on Sensors, Mesh and Ad Hoc Communications and Networks (IEEE SECON)*, 2006.
[10] K. Shashi Prabh and T. Abdelzaher. On scheduling and real-time capacity of hexagonal wireless sensor networks. *19th Euromicro Conference on Real-Time Systems, 2007. (ECRTS)*, pages 136 – 145, 2007.