EVER
ECOLOGIC VEHICLES • RENEWABLE ENERGIES
MONACO
MARCH 25-28 2010

Opportunistic Synchronization for Improving IEEE 802.15.4 MAC Performance in Chain Topologies

Ilenia Tinnirello, Laura Giarré, Francesco Mineo

Università di Palermo, Palermo 90128, Italy
E-mail: ilenia.tinnirello@tti.unipa.it, giarre@unipa.it

Abstract: *The most effective solution for energy saving in low-rate wireless sensor networks is maintaining each node in a doze state as long as possible. In order to guarantee network connectivity, the intervals at which the network sensors are turned on and off have to be coordinated. Therefore, synchronization is a very critical function for sensor networks.*

In this paper, we analyze the IEEE 802.15.4 MAC performance in sensor networks deployed in a chain topology. For this topology, critical inefficiencies can arise in case of multi-hop packet deliveries. We evaluate the impact of different synchronization schemes on the network performance, both in terms of network throughput and in terms of energy consumption. We also show how the synchronization function can be opportunistically configured in order to emulate a token-like access mechanism, able to significantly improve the network performance.

Keywords: sensor networks, medium access control, energy saving

1. Introduction

The joint efforts of the IEEE 802.15.4 task group and the Zigbee Alliance have ended up with the specification of a standard protocol stack for low-rate wireless sensor networks, which is experiencing an increasing interest and consensus by industry and academia, becoming a universal solution for low-cost low-power monitoring and control devices.

Undoubtedly, the most important features which characterized Zigbee success are the low power consumption and the real-time guarantees. Both the features are enabled by an optional synchronization functionality available at the Medium Access Control (MAC) sublayer. In fact, Zigbee includes a beacon-enabled mode, in which all nodes in the network maintain a temporal synchronization with periodic control frames called beacons. This feature allows each node to remain active during small time intervals which follow the beacon transmissions, thus guaranteeing low duty cycles and periodic slot allocations for real-time traffic. Synchronization is basically provided by the action of a central device, called Personal Area Network (PAN) coordinator, which periodically starts beacon transmissions to its neighbours. The beacons can also be propagated by other coordination nodes (called Zigbee routers) in case of multi-hop network topologies.

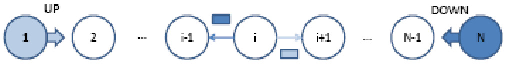

Fig. 1. Reference scenario: chain topology with bidirectional traffic flows.

Beacon propagation suffers from lacking scalability, because in absence of interaction between the MAC sub-layer and the routing layer which builds the network topology, a large amount of network resources can be wasted by beacon collisions. In fact, when periodic beacon frames are sent by multiple coordinators in a non organized fashion (i.e. without a specific schedule), they can collide either with each other or with data frames. This phenomenon is exacerbated when also data frames are generated periodically. Since data transmissions are performed in the temporal window which immediately follow the beacon transmissions, the global synchronization among the nodes can result in data loss due to the synchronized transmissions of hidden nodes (i.e. nodes which do not hear each other).

In this paper we face the problem of network synchronization, both in terms of potential benefits for energy saving and potential risks for frame collisions, when sensor nodes are deployed in a chain topology. Specifically, we consider the scenario illustrated in figure 1, where i) N nodes are equally spaced along a line topology, and ii) traffic flows are generated at the edge nodes of the line towards the opposite edge node. In this scenario, a generic node *i* of the chain has to forward frames in two opposite directions, called *up* and *down*. Since each node can hear only the previous and the next node of the chain, nodes *i-1* and *i+1* are hidden. Therefore, when node *i-1* transmits a frame to node *i*, such a frame can collide with another frame transmitted by node *i+1*. This phenomenon is in principle rare with low-rate traffic sources (if the probability to have two frame travelling simultaneously along the chain is low). However, if nodes employ low duty-cycles and wake up simultaneously, the probability to synchronize hidden node transmissions can increase significantly. Therefore, we propose a simple solution, based on a distributed scheduling of beacon transmissions, able to limit frame collisions in the network, while maintaining low duty cycles.

The rest of the paper is organized as follows. Section 2, briefly summarizes some synchronization solutions proposed so far for sensor networks. Section 3 describes the IEEE 802.15.4 basic functionalities and the iinefficiencies arising in chain topologies. Section 4 introduces our opportunistic synchronization scheme and provides some numerical results. Finally, conclusions are drawn in section 5.

2. Synchronization Solutions for Sensor Networks

Most sensor networks applications, regardless of the sensor technology, coverage capability, energy constraints, and so on, require a global clock synchronization, that is all the nodes of the network need to refer to a common notion of time. Time synchronization is also critical to sensor network efficiency and performance, since energy consumption can be reduced only keeping the radio turned off as long as possible. As reported in [1], it is necessary to design a time schedule in which all nodes turn on their radios at designated times, transmit the data they have recorded to the base station via multi-hop, and then turn off the radio again. If nodes are awake at different times, the sensor network might not be fully connected and therefore packets might not be correctly delivered to the base station. The time synchronization accuracy is related to power consumption, since it is necessary to increase the duration of the awake intervals, to compensate for synchronization errors. The time synchronization protocol itself consumes energy and need to be carefully designed in order to reduce the power saving, [2].

Time-synchronization of a wireless sensor network has two major problems. First, in a sensor network the nodes cannot communicate directly with each other but they have to do it via multi-hop. Therefore, it is not possible to choose a reference node to which all other nodes can be synchronized to. Then, because an unpredictable time delay between the clock reading in one node and its processing at the receiver node, this causes poor performance. In fact, delivery time of radio messages in WSNs are subject to random variations due to many factors, such as interference, backoff due to occupied radio channel, and scheduling of the host operating system of the motes. These delays can be magnitudes larger than the required precision of time synchronization. Different strategies has been proposed to solve these two problems. In [3] the main four global synchronization in sensor networks are described and revised: a node-based approach, a hierarchical cluster-based method, a diffusion-based method, and a fault tolerant diffusion-based method. In [1] a complete decentralized method based on consensus approach is presented. This is a fully distributed algorithm assuring robustness to node failure and to new node appearance.

Different protocols have been proposed for implementing centralized or distributed synchronization schemes in actual networks. One of the solutions, proposed by [4] is the Time Synchronization Protocol for Sensor Networks (TPSN), where the network is organized in a rooted tree and one node is the clock reference, while a spanning tree is derived from that node. In [5] a similar approach, called Floating Time Synchronization Protocol (FTSP), adds also the skew compensation and the root failure recovery. The Reference Broadcast Synchronization (RBS) consider various independent clusters, which reference nodes are synchronized in an upper level acting as gateways [6].

3. Zigbee performance over Chain Topologies

A) Overview of Zigbee

We assume that the reader is familiar with IEEE 802.15.4 MAC operation. Therefore, in this section we only summarize the MAC configuration options. The IEEE 802.15.4/ZigBee MAC protocol supports two operational modes that may be selected by the Zigbee Coordinator (ZC):

i) non beacon-enabled mode
ii) beacon-enabled mode

In the first one the MAC is simply ruled by non-slotted CSMA/CA and in the second one beacons are periodically sent by the Zigbee coordinator to synchronize nodes that are associated with it, and to identify the PAN. Hereafter, we consider the beacon-enabled mode and analyze its deployment in cluster-tree networks. In beacon-enabled mode, the ZC defines a superframe structure which is constructed basen on (1) for $0 \leq SO \leq BO \leq 14$:

$$BI = aBaseSuperFrameDuration\ 2^{BO}$$
$$SD = aBaseSuperFrameDuration\ 2^{SO} \qquad (1)$$

where the Beacon Interval (BI) defines the time between two consecutive beacon frames, the Superframe Duration (SD) defines the active portion in the BI, and is divided into 16 equally-sized time slots, during which frame transmissions are allowed. Optionally, an inactive period is defined if BI > SD.

During the inactive period, all nodes may enter in a sleep mode (to save energy). BI and SD are determined by two parameters, the Beacon Order (BO) and he Superframe Order (SO), and by the

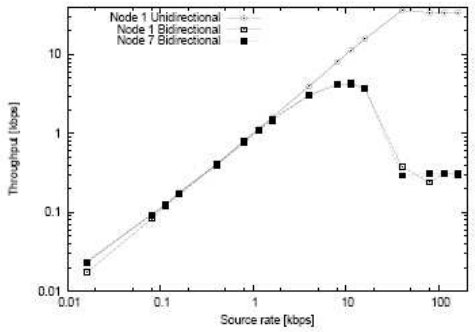

Fig. 2. Zigbee performance (no-beacon mode) over a chain topology.

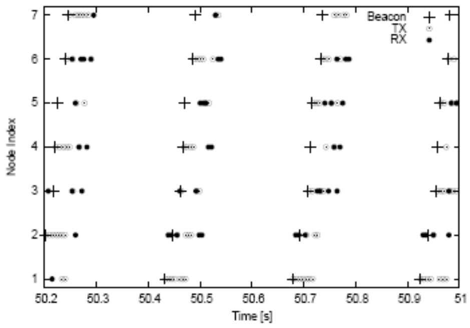

Fig. 3. Frame transmission scheduling in a synchronized chain, for a source rate of 8Kbps (BO=4, SO=2).

aBaseSuperframeDuration, denoting the minum duration of the superframe. During the SD, nodes compete for medium access using slotted CSMA/CA in the Contention Access Period (CAP). For time-sensitive applications, IEEE 802.15.4 enables the definition of a Contention-Free Period (CFP) within the SD, by the allocation of Guaranteed Time Slots (GTS). The advantage of this synchronization with periodic beacon frame transmissions from the Zigbee coordinator is that all nodes wake up and enter sleep mode at the same time. However, using this synchronization scheme in a cluster-tree network with multiple coordinators sending beacon frames, each with its own beacon interval, is a challenging problem due to beacon frame collisions.

B) Multi-hop Zigbee performance

In figure 1 each node is identified by a number, which corresponds to the node position along the chain, from node 1 to node N. We assume that the distance between two consecutive nodes of the chain is set approximately to a constant value. Note that these assumptions can be very realistic whenever the physical location of the sensors is planned a priori. Since our network topology only includes the traffic sources (edge nodes) and the intermediate relayers, we assume that the transmission range is equal to one hop only. Unless otherwise specified, we also assume that transmission range and carrier sense range coincide. Consider first the case in which only node 1 generates traffic towards node N. This means that all the intermediate nods act as sequential relayers. The data packets transmitted by node 1 are received by node 2, which forwards the packets to node 3, and so on, until the destination N is reached. In a single-hop network, the presence of a single traffic source assures that there is no collision on the channel and the data reception rate is the maximum possible Sx. In such a case, the channel wastes are only due to the data frame overheads (both at the PHY and MAC layer) and to the protocol overheads given by the backoff expiration times. For example, for a payload size of 100 bytes and a data rate of 250kbps, the single hop maximum throughput achievable is about 120kbps [7].

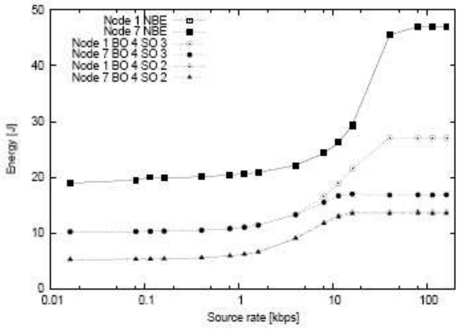

Fig. 4. Comparison between energy consumption in no-beacon and beacon-enabled mode.

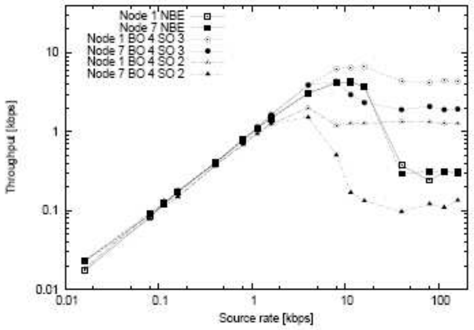

Fig. 5. Comparison between throughput performance in no-beacon and

In a multiple-hop network, even in the presence of a single traffic source, the relayer nodes may contend for the channel access and may collide because of simultaneous transmissions and carrier sense failures. For example, in the assumption that the carrier sense range is equal to one hop, during the forwarding of different data packets, transmissions by node *i−1* and node *i+1* could overlap because these nodes cannot hear each other. This overlapping results in the failure of node i reception (i.e. the failure of node i−1 transmission), because node *i* is in the transmission range of both node *i−1* and node *i + 1*.

In order to avoid this hidden transmissions and to maximize the number of simultaneous deliveries along a node chain, an ideal transmission scheduling should cyclically synchronize the transmissions of node (1, 4, 7,.), (2, 5, 8,.), (3, 6, 9,-). In such an ideal case, the maximum achievable throughput of the chain results Sx/3, since the traffic source has to wait for two relayer transmissions before proceeding with the next data packet. Obviously, hidden transmissions are not a serious concern in case of low-rate traffic. In fact, when node *i − 1* receives a new packet, node *i + 1* has already transmitted its pending frame and the collision probability is negligible. Thus, while the intermediate node queues are maintained empty, frame sequential delivery acts as token passing.

Consider now the case of bidirectional traffic. When both node 1 and node N generate traffic, hidden transmissions can occur even in presence of low-rate traffic. A visualization of these considerations can be drawn from figure 2. In the figure, we report NS-2 based simulation results of the throughput experienced by the edge nodes of a network made up of 7 nodes. From the figure, it is evident that, as the load offered by the edge nodes increases, the experienced throughput decreases to a very low value (namely, only 300bit/s). Conversely, in the case of unidirectional traffic from node 1 to node 7, the maximum sustainable throughput of the chain is about Sx/3, i.e. slightly less than 40kbps.

C. Multi-hop Performance in Beacon-Enabled Mode

In order to quantify the impact of synchronization on Zigbee performance in chain topologies, we repeated previous simulations in case of beacon-enabled networks, see [8]. We assume that the first node of the chain acts as a network coordinator. In these conditions, at each beacon interval, each node *i>1* is synchronized with the previous node (which acts as a parent) and plays the role of parent for the next node of the chain (which acts as a child). In other words, each node has to wake up for receiving the beacons of the previous node and has to periodically schedule its beacon transmission for the next node. The standard does not specify the rules for starting the beacon propagations. Therefore, in most current implementations, devices acting both as a child and as a parent, schedule their beacons right after the association to their coordinator. This implies that the time offset between beacons sent by consecutive nodes is a random value and can also be small.

Figure 3 visualizes the temporal sequence of the frame transmissions along the chain in a temporal window between 50.2 s and 51 s. The figure has been obtained for a source rate of 8 kbps at node 0 and node 6, and for BO=4 and SO=2. Each node is indicated by its identifier on the y-axis, and the node activity is indicated by different point types, representing beacon transmissions, data frame receptions and data frame transmissions. From the figure, it is evident that each node is active only for 1/4 of the beacon interval. Collisions occur whenever two nodes whose distance is equal to 2 hops transmit simultaneously. For example, at 50.44 s, both nodes 2 and 4 transmit and node 3 has an activity hole because it is not able to decode its data frames. Beacons transmitted by consecutive nodes are shifted randomly.

Using beacons have two different advantages on network performance. On one side, there is the obvious advantage of energy saving, by keeping the radio on during a fraction of the beacon interval. On the other side, synchronizing transmissions along the chain reduces the frame delivery delay (or improves the throughput). These phenomena are quantified in figures 5 and 4, which compare the energy consumption and the network throughput experienced in case of no-beacon mode and beacon-enabled mode, for BO=4 and SO=3 and BO=4 and SO=2. The figures also show that there is an asymmetry between the two traffic flows, (up and down), due to the fact that node 1 acts as a coordinator (i.e. starts the beacon propagation).

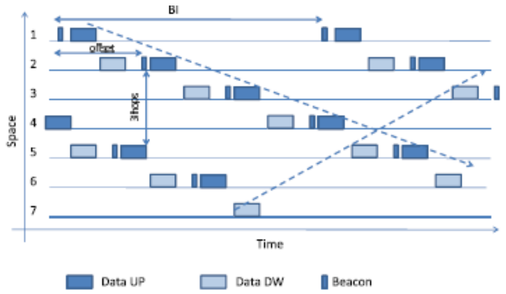

Fig. 6. Token-like polling scheme.

4 Opportunistic Synchronization in Chain Topologies

In this section, we analyze the possibility of exploiting a distributed scheduling mechanism of beacon propagations in order to further improve the network performance in beacon mode. We assume that node are aware of network topology and that node 1 acts as a coordinator. The opportunistic frame transmissions are represented in a spatial-temporal matrix. Specifically, figure 6 plots the frames transmitted simultaneously in the same column, and the frames transmitted by a given node in the same row (similarly to figure 3). The chain is made up of seven nodes, with node 1 transmitting traffic towards node 7 (up flow) and node 7 transmitting traffic towards node 0 (down flow). Frame transmissions are scheduled as follows. Node 1 sends its beacon, followed by a data frame to node 2. Node 2 wakes up at node 1 beacon transmission and sends back a data frame to node 1. After a temporal offset from node 1 beacon, node 2 starts its beacon transmission. The frame sequence beacon, up data, and down data is then repeated between node 2 and node 3, node 3 and node 4, and so on, by sequentially passing the coordinator role along the chain as a token. In the meanwhile, at regular beacon intervals, new beacons are scheduled by node 1 and the frame sequence just described from node 1 to node 7 is repeated. By opportunistically setting the beacon offsets and the beacon intervals, it is possible to regulate the spatial reuse along the chain in order to have simultaneous transmissions along the chain which do not interfere. Whenever the beacons sent by node 1 are spaced by at least the time interval required for starting the beacon transmission of node 4, simultaneous transmissions have a distance of at least three hops, and no frame collides. Obviously, for carrier sense ranges greater than one hop, the beacon temporal offset has to be increased accordingly [9]. Given the desired spatial reuse, the timing of the beacon transmissions at node 1 is not very critical. Slightly late transmissions reduce the channel efficiency without causing collisions. We

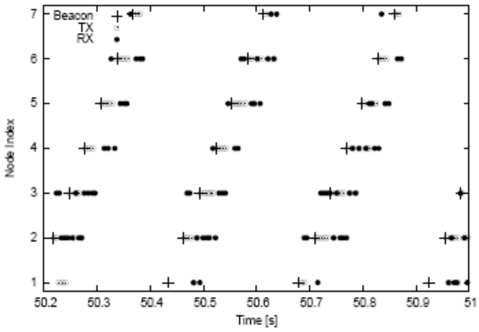

Fig. 7. Opportunistic frame transmission scheduling, for a source rate of 8Kbps (BO=4, SO=2).

do not discuss here the problem of setting and/or correcting the beacon offset value. We just assume that the setting is based on the carrier range estimation carried out during the set-up phase. Note that uplink forwarding is quicker than the downlink one, because the uplink frame reaches the destination in a single token passing cycle from node 1 to node N. Conversely, the delivery of the downlink packet requires N − 1 token cycles, because, during the same token cycle, each node of the chain transmits different downlink frames.

A. Performance Evaluation

We implemented our opportunistic beacon scheduling scheme in the NS2 simulator platform. Figure 7 visualizes the temporal sequence of the frame transmissions along the chain in a temporal window between 50.2 s and 51 s, in the same experiment of figure 3 (i.e. a source rate of 8 kbps and BO=4 and SO=2). We can see how beacons are regularly shifted from one node to the next one and how this synchronization significantly reduces the network collisions (i.e. the activity holes during the superframe duration). Transmissions (white circles) and receptions (black circles) tend to be performed in a batch. Figure 8 quantifies our previous considerations, for a time offset equal to about 1/4 of the activity period. While the standard beacon-enabled mode gives a saturation throughput of about 1 kbps for BO=4 and SO=2 and 6.5 kbps for BO=4 and SO=3, using a regular time offset among consecutive nodes, such a throughput is increased respectively to 5.5 kbps and 12.5 kbps (i.e. it allows to approximately double the throughput performance). A further advantage is obtained in terms of energy saving, due to the reduced number of collisions. We omit such a quantification for space reasons.

5 Conclusions

In this paper, we propose a time-division approach for scheduling beacon transmissions in a Zigbee network deployed in a chain topology. We intuitively justify the relationship between temporal offsets between beacon transmissions performed by different nodes and network spatial reuse. By exploiting this relationship, we define a distributed beacon scheduling policy able to improve the performance of the IEEE 802.15.4 MAC, avoiding collisions and enhancing the energy saving. Simulations and comparisons with standard solutions have been presented showing the increased performance.

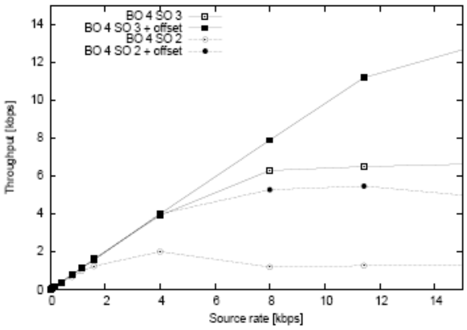

Fig. 8. Token-like polling scheme: effects of timing errors and frame errors.

Acknowledgement

This work has been partially supported by the European FP7 Project BEYWATCH.

References

1] L. Schenato, G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In proceeding of the IEEE CDC, San Diego, CA, USA, December 2007.

[2] Stephen F. Bush, Low-Energy Sensor Network Time Synchronization as an Emergent Property . In Proc. of Computer Communications and Networks IEEE conference, 93-98, 2005.

[3] Qun Li and Daniela Rus. Global Clock Synchronization in Sensor Networks. IEEE Trans. On Computers, VOL. 55, NO. 2, FEBRUARY 2006.

[4] S. Ganeriwal, R. Kumar, and M. Srivastava. Timing sync protocol for sensor networks. In Proceedings of the first international conference on Embedded networked sensor systems (SenSys03), 2003.

[5] M. Maroti, B. Kusy, G. Simon, and Akos Ldeczi. The flooding time synchronization protocol. In SenSys 04: Proceedings of the 2nd international conference on Embedded networked sensor systems, pages 3949, 2004.

[6] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. Proc. of the 5th symposium on Operating systems design and implementation OSDI 2002, pp. 147-163, 2002.

[7] B. Latre, et. al. Throughput and Delay Analysis of Unslotted IEEE 802.15.4. Journal of Networks, Vol. 1, No. 1, May 2006, pp. 20-28.

[8] Anis Kouba, Andr Cunha1, Mrio Alves. A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks. in Proc. 19th Euromicro Conference on Real-Time Systems (ECRTS 2007), July 2007.

[9] I. Tinnirello, L. Scalia, F. Campoccia. Improving IEEE 802.11 Performance in Chain Topologies through Distributed Polling and Network Coding. Proc. of IEEE ICC 2009, Dresden, June 2009, pp.1-6.