This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

A Non-beaconing ZigBee Network Implementation and Performance Study

Magnus Armholt
Institute of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: magnus.armholt@tut.fi

Sakari Junnila
Institute of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: sakari.junnila@tut.fi

Irek Defee
Institute of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: irek.defee@tut.fi

Abstract— **In this paper we present our implementation of a layered ZigBee network and the performance test results using non-beaconing networks. Adding the implemented network layer to an existing IEEE 802.15.4 stack did not affect the throughput in the network. It is also concluded that with knowledge of the network structure an extra added delay between sending packets can reduce the number of packets lost due to channel access failure without decrease in throughput.**

I. Introduction

The ZigBee standard [1] is a standard for low-power consuming wireless devices, operating in the industrial, scientific and medical (ISM) radio bands, 868 MHz in Europe 915 MHz in the USA and 2,4 GHz in most countries of the world. The standard defines two types of devices; Full function device (FFD) and Reduced function device (RFD). An FFD usually operates as a coordinator or a router but can also act as an end-device. A RFD on the other hand is designed with power consumption in mind and can thus only operate as an end-device. The FFD is usually connected to a mains-power supply while the RFD is typically battery operated. A ZigBee network can be either beacon-enabled or non-beacon-enabled. In a non-beacon-enabled network all packets are sent using unslotted CSMA-CA. The supported network types are; star, mesh and cluster-tree topology. In star topology, the network in controlled by a single network coordinator. The mesh and cluster-tree networks extend the network onwards from the coordinator using router devices. Mesh, e.g., peer-to-peer networks provide multiple path options, which enhance reliability/scalability in the network. Cluster-tree networks utilize a hybrid star/mesh topology, which gives better support for battery powered nodes in the network as only the router nodes are responsible for data relay in the network. The routers in cluster-tree networks utilize a hierarchical routing strategy, and they may also employ beacon oriented communication, while mesh networks allow peer-to-peer communication and shall not emit beacons.

The ZigBee stack architecture, see Fig. 1, is inspired by the open systems interconnection (OSI) seven-layer model [2] and is divided into four distinct layers; the physical layer (PHY), the medium access control layer (MAC), the network layer (NWK) and the application layer (APP). I.e., the three lowest layers are implemented as such, and the APP layer combines the main functionality of the four higher layers. The two lower layers, PHY and MAC, are defined by the IEEE 802.15.4 standard [3]. The features and functionality provided by the MAC-sublayer are: network association and disassociation, acknowledgement frame delivery, frame validation, channel access (using CSMA-CA), beacon management (optional), and guaranteed time slot (GTS) mechanism for high-priority communications. The GTS mechanism is not used by the ZigBee NWK layer. In addition, the MAC sublayer provides support for implementing security mechanisms. Building on this base, the network layer adds functionality for discovering and maintaining routes, starting a network, functions for joining or leaving a network, and the ability for the coordinator to assign short address' to devices joining the network. The network layer can also secure the transmissions and synchronize devices within the network, but security was not implemented by us and the synchronization could not be fully implemented without MAC beaconing support. Finally, the ZigBee application layer adds binding to the device, meaning that more than one application can use the same device.

In especially science and education a layered implementation can be of interest due to the fact that it makes it easy to compare different implementations. This work focuses on implementing the ZigBee network layer from a layered point of view. Accessing the lower layers is done through the public functions described in the IEEE 802.15.4 standard and the interface to the application layer is constructed according to

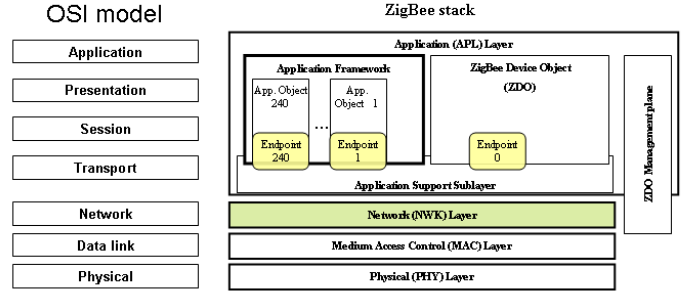

Fig. 1. The OSI model and the ZigBee stack.

1-4244-0353-7/07/$25.00 ©2007 IEEE

the public functions in the ZigBee specification. The IEEE 802.15.4 stack used in this project does not support sending out beacons and processing other received beacons, hence the implementation is only tested for non-beaconing networks.

A ZigBee stack can be bought from several companies and a free stack is available from Microchip Technology Inc. for PIC microcontrollers. Also, Dr. Robert Reese [4] from Mississippi State University has developed a platform independent subset of the ZigBee stack, which is available for free for research purposes. This stack was not available at the start of this project. Other published research work in this area has mostly been based on simulations and mainly focused on the IEEE 802.15.4 MAC. Lu et al. [5] studied the performance of IEEE 802.15.4 in the NS-2 simulator and Lee [6] performed a indoor measurement study of the IEEE 802.15.4 performance. Petrova et al. [7] compared NS-2 simulations with indoor and outdoor IEEE 802.15.4 measurements. We could not find any existing studies incorporating the network layer.

In Section 2 we describe the implementation specific choices and issues. Section 3 is dedicated to the experimental setup and in Section 4 the results are presented. Finally, in Section 5 the conclusions are given.

II. Implementation

The hardware used in this project consists of three ZigBee radio boards; one Chipcon CC2420 v.1.1. and two Chipcon CC2420 v.1.2 development boards. To record the network traffic a packet sniffer was used, using Chipcon CC2400 v.2.0 evaluation board with a Chipcon CC2420 v.1.0 RF-module connected to it.

The implementation of the network layer is written in C and compiled with avr-gcc v.3.4.6. The underlying IEEE 802.15.4 MAC used was the Chipcon MAC stack v.1.3.0.

A. Timer issues

In the ZigBee standard it is said that the network layer shall request retransmissions of route requests after a certain time period. Also other parts of the network layer need to perform actions after some time has elapsed. This has led to the decision to implement a scheduler in the network layer. However, the microcontroller on the CC2420DB has four timers of which only two are 16 bit, and other two 8 bit. The IEEE 802.15.4 stack already uses one of these 16 bit timers for MAC timing. We felt that we should leave the other 16 bit timer for possible applications to use and opted to use the MAC timer for also the NWK stack. Because of this, the strict layered structure of this project was abandoned. The timer interrupt calls the timer interrupt routine of the MAC layer, which after processing the MAC functions, calls the network layer interrupt scheduler.

B. Reuse of network addresses

The ZigBee standard specifies two address assignment mechanisms; the tree address allocation and the higher layer address assignment. A ZigBee network address is 16-bits so the address space is finite and reuse of addresses is needed. When higher layer address assignment is used in this project, also reuse of address has been left to the higher layers.

For tree address allocation the ZigBee standard specifies two different equations for assigning addresses to router-capable children and end-device children. Addresses for router-capable children were assigned according to equation 1 where A_{parent} is the address of the current device, $Cskip(d)$ is a depth specific skip value, d is the depth of the current device, $1 \leq n_{router} \leq Rm$, and Rm is the maximum number of router-capable children a device may have.

$$A_{n_{router}} = A_{parent} + 1 + Cskip(d) \cdot (n_{router} - 1) \qquad (1)$$

Network addresses are assigned to end devices according to equation 2, where $1 \leq n_{enddevice} \leq (Cm - Rm)$, and Cm is the maximum number of children a device may have.

$$A_{n_{enddevice}} = A_{parent} + Cskip(d) \cdot Rm + n_{enddevice} \qquad (2)$$

The reuse of tree allocated address is implemented using a bitmap representing the n_{router} and another bitmap representing the $n_{enddevice}$. When a device leaves a network in such a way that the address is allowed to be reused then the bit in the bitmap representing the n_{router} or $n_{enddevice}$ is set in the corresponding bitmap. When assigning a new address to a joining device the bit map is checked and if a bit is set then the corresponding n_{router} or $n_{enddevice}$ is used instead of increasing the internally kept counter.

III. Experimental setup

A non-beaconing network was formed on channel 11, which is the first of the 16 2.4 GHz ISM-band channels, using three CC2420DBs; one coordinator having two children, from which one is acting as an end-device and the other one as a router-capable child. In all tests except test A, the packets were sent from the end-device via the coordinator to the router-capable child. The addressing mode was the 16-bit network addressing. The tests were conducted in office environment with a distance of about 40 cm between the devices. Throughout this paper, when talking about packet size, this refers to the actual data payload of the packet, discarding the headers and the footers. Also the throughput calculated is based on the actual data moved from point a to point b, saying that the size of the original data sent is divided with the total time, including the time for retransmissions.

A. Acknowledgment time

The first test conducted was to determine the acknowledgment time, meaning the time between packet sent and acknowledgment received. A non-beaconing network was formed and the device joined the network. 40 packets were sent from the device to the coordinator and the acknowledgment time was observed. The test was then repeated sending 40 packets from the coordinator to the device.

B. Throughput in burst mode

This test was conducted to study the effects of the network layer on the throughput. Throughout this report a burst mode means that the next packet is requested to be sent as soon as the acknowledgment from the previous packet is received. Once the non-beaconing network was up, an initial packet was sent to let the coordinator find a valid route to the router-capable device. When this was done, 40 packets were sent in burst mode with the router-capable device as the destination. The test was repeated 10 times and the traffic recorded with the packet sniffer. Retransmissions were utilized but if any packet was still lost (due to channel access failure) the test was discarded and repeated. This is because the occurrence of channel access failures was tested in the test presented in the next subsection.

To see the effects of the network layer, the same test was repeated using only the IEEE 802.15.4 stack. To make the comparison, all packets sent to the coordinator were requested to be sent to the router-capable device as soon as they were received at the network layer. The added functionality of the network layer over MAC layer in this test was to recognize the intended receiver of the packet and look up routing information for the intended receiver.

C. Packet loss due to channel access failure

When using unslotted CSMA-CA the transmission of a packet is first delayed with an initial random backoff time. The physical layer senses the channel and if the channel is busy an random backoff period is waited before the channel is sensed again. This procedure is repeated a couple of times but if the channel is busy for a longer period the MAC layer will drop the packet and indicate a channel access failure to the network layer.

This test was conducted to determine how often this occurs. A non-beaconing network was formed and an initial packet was sent to let the coordinator find a valid route to the router-capable child. Packets were sent in burst mode and the test was repeated 10 times.

D. Effects of steady flow

Reliability in ZigBee is implemented by retransmissions. However, retransmissions do not solve the problem with packets lost due to channel access failure. This test was conducted to try to find a waiting period between packets where packets do not collide and what throughput this will lead to.

To form a steady flow, a time was waited between the reception of the acknowledgment (for the previous packet) and the request for the next packet to be sent. A non-beaconing network was set up and an initial packet was sent to let the coordinator find a valid route to the router-capable child. Since the aim was to observe a loss of packets due to channel access failure the biggest packet size possible was used since this will occupy the channel the most. 70 packets were sent and the test was repeated 10 times.

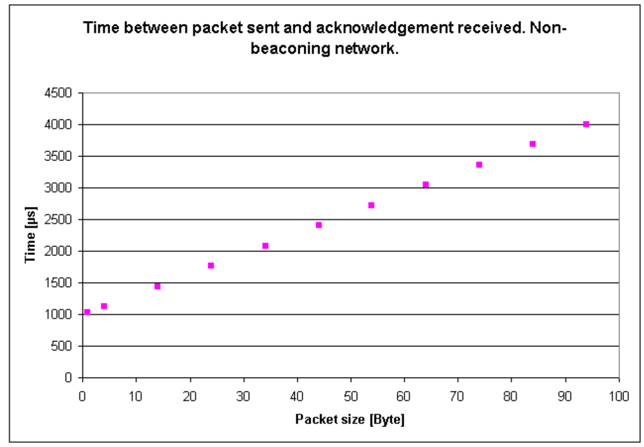

Fig. 2. Acknowledgment time in non-beaconing network.

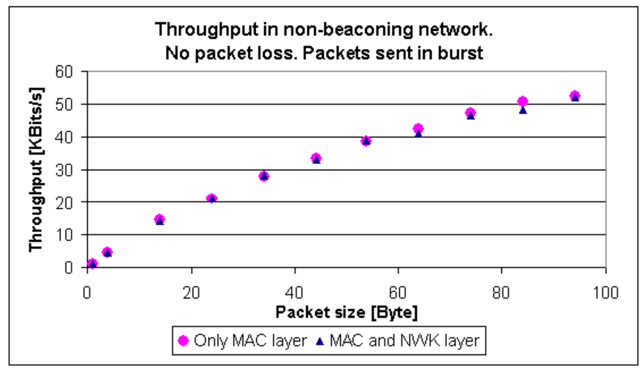

Fig. 3. Throughput in non-beaconing network, packets sent in burst.

IV. Results

The time used in calculations and reported in the results from experiment IV-A is the time recorded by the packet sniffer. In the case of experiment IV-A this means the time between the packet sniffer received the packet and the acknowledgment.

A. Measuring acknowledgment time

Measuring the acknowledgment time shows as expected an relationship between the packet size and the acknowledgment time. The results are visualized in Fig. 2. Acknowledgment is sent as soon as the packet has been received, before it is passed to the upper layers. Therefore, there is no difference in acknowledgment time when using more layers in the stack. Also worth noticing is that acknowledgments are sent without the use of CSMA-CA.

B. Throughput in burst mode

Figure 3 shows the results of the measurements. From the results it can be seen that there is no difference in throughput if only the IEEE 802.15.4 stack is used or if also the network layer is added.

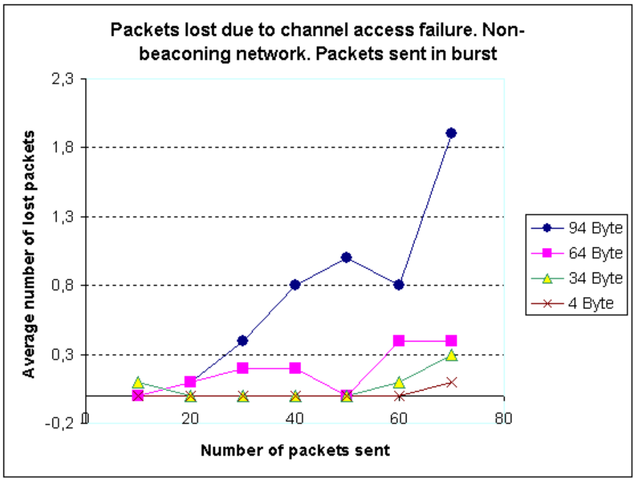

Fig. 4. Packet loss in non-beaconing network, packets sent in burst.

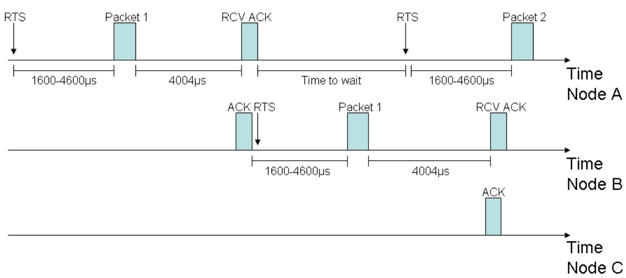

Fig. 5. Transmission time lines for node A, B and C.

C. Packet loss due to channel access failure

The results for the test are shown in Fig. 4. As expected, the bigger the packet size gets, the more packets are lost due to channel access failure. The small number of trials (10) causes fluctuations and dips in the curves, but the trend of the curves can be seen already from these measurements.

D. Effects of steady flow

Figures 6 and 7 illustrate the results. Analysing Fig. 6 a steady trend until 3000 μs can be seen, then an increase at 4000 μs and from 5000 μs a linear decrease.

This can be explained by looking at the acknowledgment time and the time between request to send and the actual transmission. Fig. 5 illustrates the transmissions and times for one packet, where node A is the source (end-device child), node B is the relaying node (coordinator) and node C is the destination (router-capable child). During the experiments the time between request to send (RTS) and actual transmission was observed to be random between 1600 μs and 4600 μs. These times depend on the MAC implementation of the CSMA-CA algorithm, which includes an initial random delay. Looking at Fig. 5 and the throughput in Fig. 6 it can be noted that the peak in throughput at 4000 μs is referred to the fact that 4000 μs is the time for an acknowledgment.

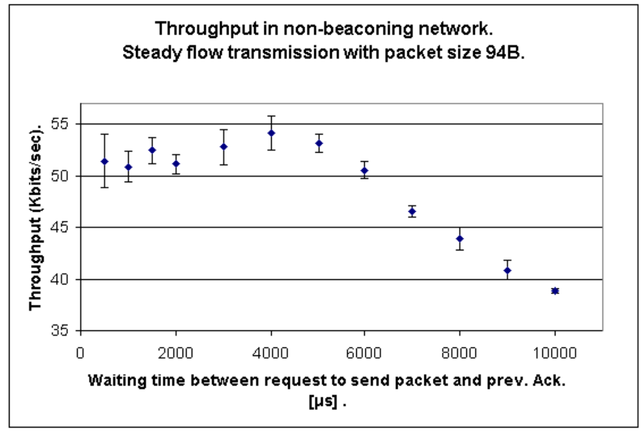

Fig. 6. Throughput in non-beaconing network, steady flow.

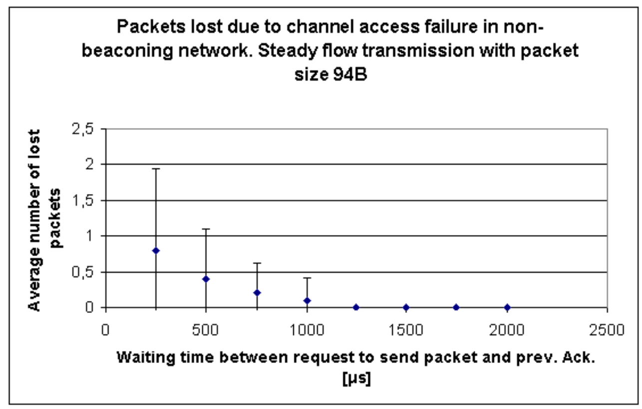

Fig. 7. Packet loss in non-beaconing network, steady flow.

When the time to wait is equal to the acknowledgment time then the next packet will be tried to be sent immediately after the acknowledgment is sent between the destination and the relaying node. When the time to wait is smaller than the acknowledgment time, then the next packet and the acknowledgment will both be trying to use the channel, leading to a backoff in the MAC layers CSMA-CA algorithm. After 5000 μs the difference between the acknowledgment and the exceeding time to wait is greater than the backoff time and hence the throughput gets lower.

Channel access failure occurs when the channel is busy for a longer time. When we add an extra delay between request to send packets, a time gap in between transmissions is created and any node waiting to send a packet can access the channel. Observing the number of recorded packets lost due to channel access failure, as visualized in Fig. 7, it can be noted that when sending as little as 70 packets a packet loss can be observed with up to 1200 μs waiting time. Packets are probably lost above this limit as well but they cannot be observed with this low amount of packets sent. In the scenario with only three devices a minimum worst case waiting time can be calculated by assuming that the RTS time for the relaying node of packet

1 is the maximum observed, 4600 μs, and the RTS time for the source node of packet 2 is the minimum observed, 1600 μs. Adding this time difference with the acknowledgment time, 4004 μs, gives a minimum worst case waiting time, 7004 μs. In the average case the RTS time is the same and they even out each other. Hence the peak of throughput when the waiting time is equal to the acknowledgment time. If the scenario is extended to four devices an another 3000 μs needs to be added to the waiting time to be certain that the next packet does not collide with the last acknowledgment. This scenario assumes that all devices are within reception range of each other.

V. Conclusions

This paper presented experiences made in implementing a layered ZigBee network layer and performance test results of this implementation in non-beaconing networks. Due to timer issues the strict layered implementation had to be compromised. The test results do not show significant impact on throughput when adding the network layer on top of the IEEE 802.15.4 stack. It can also be concluded that with knowledge of the network structure, adding a waiting period between sending packets can reduce the probability of channel access failure without a decrease in throughput.

VI. Discussion

What to do when a channel access failure occurs in a relaying device is in the ZigBee standard left to the implementer. When it occurs at the source of a packet the error is sent up to the application layer where a decision can be made. In a relaying node the error message can not be sent to the application layer since it is not the source of the frame. When the MAC layer has tried to send the packet and is about to indicate channel access failure, it drops the packet from the transmission queue and indicates an error to the network layer. To avoid losing the packet totally the network layer could implement an packet buffer having the same size as the MAC layer transmission queue and thus have a possibility to request the frame to be sent again. This however puts the frame at the end of the transmission queue which would lead to packets being transmitted out of order. Another solution would be to implement a way for the application layer to request missing packets, assuming that the data was fragmented over more than one packet due to packet size limitations.

A. Future work

Future work includes reducing the size of the network layer implementation and adding an application layer. Also, the behavior when a channel access failure occurs needs to be examined and evaluated in order to get reliable transmission.

References

[1] Z. Alliance, *ZigBee Specification*, 1st ed., ZigBee Standards Organisation, http://www.zigbee.org, June 2005.

[2] H. Zimmermann, "The OSI Reference Model - The ISO model of architecture for Open Systems Interconnection," *IEEE Transactions on communication*, vol. 28, pp. 425–432, April 1980.

[3] I. C. Society, *IEEE Std 802.15.4-2003*, The Institute of Electrical and Electronics Engineers, inc, http://www.ieee.org, October 2003.

[4] R. Reese, http://www.ece.msstate.edu/˜reese.

[5] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proc. IEEE Int Conf on Performance, Computing, and Communications (PCCC 2004), Los Angeles, CA, USA*, 2004, pp. 701–706.

[6] J.-S. Lee, "An experiment on performance study of IEEE 802.15.4 wireless networks," in *Proc. IEEE Int Conf on Emerging Technologies and Factory Automation (ETFA 2005), Catania, Italy*, vol. 2, Sept 2005, pp. 451–458.

[7] M. Petrova, J. Riihijärvi, P. Mähönen, and S. Labella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2006), Las Vegas, NV, USA*, vol. 1, April 2006, pp. 487–492.