Performance Evaluation of an Indoor Localization Protocol in a 802.15.4 Sensor Network

Jorge Juan Robles*, Enrique Gago Muñoz**, Laura de la Cuesta*** and Ralf Lehnert*
Chair for Telecommunications
Technische Universität Dresden, Germany
*{robles|lehnert}@ifn.et.tu-dresden.de
**engamu@hotmail.com
***lcueiba@gmail.com

***Abstract*—Nowadays there are many indoor localization systems, which use a sensor network as general platform. In this work, the protocol for localization ”Highly Configurable Protocol” (HCP) [1] was implemented in a 802.15.4 test-bed forming a tree topology.**

Our goal was to empirically investigate how the limitations of a IEEE 802.15.4 sensor network affect the performance of HCP in the localization process, principally in the transmission of localization information across the network.

The obtained results allowed to identify not only the causes of packet losses in the network, but also possible improvements to increase the efficiency of HCP and the scalability of the localization system.

I. Introduction

An indoor wireless sensor network enables the creation of multiple applications like environmental monitoring, access control as well as ambient assisted living solutions. In many of these applications, it is necessary to estimate the coordinates of a sensor node inside the building requiring the implementation of a localization mechanism.

The design of localization systems in sensor networks is a challenging issue due to the fact that the sensor nodes are very limited in terms of energy consumption, transmission range, memory capacity, data rate and processing power. Furthermore, the localization systems require the transmission of localization information and signaling through the network. This can be a problem, if several nodes need to know their positions at the same time, principally due to the capacity limitations of the network. Thus, the network protocol plays an important role in the localization systems, given the fact that it defines the rules and the way in which the nodes communicate during the localization process.

In general, the localization systems exploit inter-node measurements to estimate the position. Most of the transceivers used in sensor nodes are able to measure the signal strength of the received signal without the inclusion of additional hardware. Thus, the localization systems based on Received Signal Strength Indicator (RSSI) have become very popular, principally due to its low complexity and low cost. However, it has to be noted, that the big disadvantage of RSSI-based system is its low position accuracy ($\sim$3m - 6m) [2] compared to other approaches, e.g. the Time of Arrival (TOA)-based methods, which can achieve an accuracy level within a few centimeters [2].

This paper evaluates the performance of the protocol HCP designed for RSSI-based localization systems, which is implemented in a beaconless 802.15.4 sensor network [3].

978-1-4673-1954-6/12/$31.00 ©2012 IEEE

II. Preliminaries

A. The reference node

The sensor node used in this project is the RCB230 [4], which contains the 8-bit microcontroller Atmega1281 [5] and the 802.15.4-compliant transceiver AT86RF230 [6] operating at 2.4Ghz.

Concerning the software stack, the 802.15.4 libraries of Atmel [7] was used as basis. On this stack, the network (NWK) and the application (APP) layers were defined. Fig. 1 shows the interaction between the different layers inside a generic sensor node. The protocol HCP was implemented within the APP layer, which can communicate with the NWK layer to transmit packets through the network. HCP can also directly communicate with the 802.15.4 layers (PHY and MAC), e.g. to configure its internal parameters like the maximum number of retransmissions.

In general, the communication between layers is done by using ”primitives”. For instance, if the NWK layer needs to transmit a packet, it sends the primitive ”transmission request” to the MAC layer. Once the packet is successfully transmitted, the MAC layer sends a primitive of confirmation to the NWK layer informing about its success.

In the implemented localization system, the nodes can operate as coordinator, anchors (ANs) or Mobile Nodes (MNs). The coordinator is connected to a central computer. The central computer is in charge of gathering of localization information for the position estimation and the configuration of the network. The ANs are nodes with known positions. They are fixed in the building and can be powered externally. On the contrary, the MNs are end devices that use batteries. The localization system is used to estimate the coordinates of the MNs.

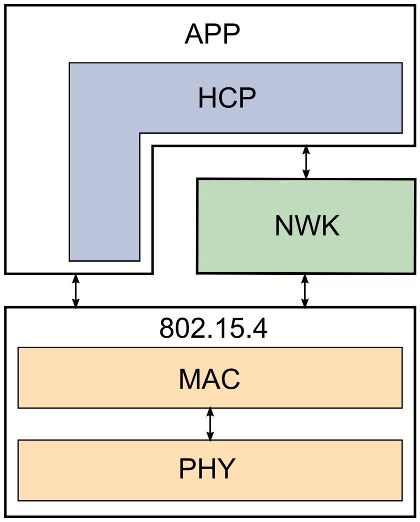

Fig. 1. Functional layers of a generic sensor node.

B. The standard IEEE 802.15.4

As indicated, the standard IEEE 802.15.4 [3] specifies the PHY and MAC layer of the communication protocol. The standard proposes the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in the MAC layer to minimize collisions between transmissions.

When the MAC layer receives a transmission request from the upper layer, the associated frame is stored in a internal queue. When the MAC layer is ready to transmit the frame, it starts generating a random delay (backoff delay) between 0 and $2^{(BE-1)}$, where BE is the Backoff Exponent. After the backoff delay, the node performs a Channel Clear Assessment (CCA), where the occupancy of the channel is analyzed. If the channel is free, the frame is transmitted and the MAC layer sends a successful confirmation to the upper layer (if a MAC acknowledgment is expected, the node has to wait for this frame before sending the confirmation primitive).

In case the channel is busy at the CCA, BE is incremented and a new random delay is generated. Then, the CCA detection is carried out again. This process can be repeated a certain number of times and is limited by the parameter *macMaxCSMABackoffs*, which is 4 by default in the standard. If the node cannot transmit the frame because the channel was busy during all attempts, then the MAC layer sends a Channel Access Failure (CAF) to the upper layer informing it about the problem. The upper layer can either discard the packet or try a retransmission by sending a new transmission request to the MAC layer.

The MAC layer can also be configured to transmit acknowledgments (ACKs) after the reception of a frame. The ACKs are very short frames without payload and they are transmitted without performing CSMA/CA.

In case the MAC is configured to receive ACKs, the MAC layer sends a primitive of successful confirmation to the upper layer, once it receives the ACK from the other side. If the ACK does not arrive, the MAC automatically retransmits the same frame and waits for an ACK again. The number of MAC retransmissions is limited by the parameter *macMaxFrameRetries* (3 by default). The MAC sends the error ”NO_ACK” to the upper layer if the transmitter did not receive any ACK after its retransmissions.

C. The addressing scheme and the network layer

The ANs are organized forming a tree topology, where the highest hierarchy level is given by the coordinator. The multi-hop communication between the ANs is performed by a routing algorithm in the network layer.

Before explaining the implemented routing algorithm, it is necessary to describe how the network is established and the addressing scheme that is used. Both, the addressing scheme and routing algorithm are taken from previous works [9] [10] and are based on [8] [12]

When an AN is switched on, it tries to join to the network by associating with an AN (or the coordinator) that already forms part of the network. If the association process is successful, the new node (child node) receives a set of addresses from its parent node. The first address of this set is recognized by the child node as its own short address. This short address (16 bits) identifies the AN within the network and will be used by the routing algorithm.

The rest of the addresses (from the assigned set of addresses) can be given by the child node in case new ANs want to associate with it. Thus, this child AN can become a parent node of other ANs repeating the described process. Each parent AN stores the assigned set of addresses of its children in its memory. Furthermore, the ANs store the short address of their parent node. In Fig.2 the address assignment is illustrated as an example in a small network. Here, the coordinator has the set of addresses between 0 and 100, where 0 is its own short address. The coordinator has given the set between 1 and 50 to the first of its children. The second child node received the set between 51 and 100. The AN with short address 1, divided its set into two subsets. These subsets were given to its two children.

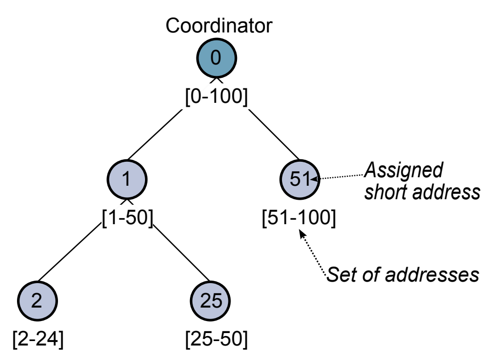

Fig. 2. Addressing scheme for the anchors.

In the implemented routing algorithm, the inherent relationship between the addresses of parent nodes and children is exploited to decide the next AN that routes the packet. Thus, when a packet arrives to an AN, it analyzes the destination

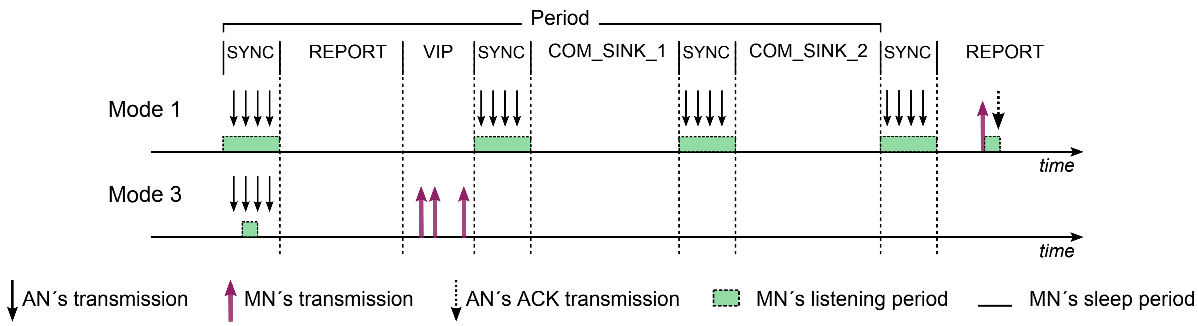

Fig. 3. MN's operation in Mode 1 and Mode 3.

address of the packet and checks if the destination address belongs to any set of addresses of its children. If so, the AN sends this packet to the corresponding child AN. If not, the packet is sent to its parent node. The simplicity of this mechanism makes this routing algorithm attractive for limited sensor nodes.

In our implementation, different parameters can be configured in the routing algorithm, like the number of NWK retransmissions to lower layers (e.g. in case the MAC layer informs about errors), the activation or not of ACKs at NWK level, etc.

The MNs are end devices that are not able to route packets. Therefore, they do not have a routing algorithm implemented in the NWK layer.

As described, a short address can be assigned during the association process. But this requires the transmission of several packets between parent and child. In order to avoid this signaling, the MNs do not use short addresses. On the contrary, an MN is identified by its long address (64 bits) within the network. The long address is a unique address originally provided by the manufacturer and saved in the non-volatile memory of the node.

In the network, each MN has a specific AN as reference. At the central computer, there is a configuration list that maps a MN's address with an AN's address of the network. In case the central computer needs to transmit some data to a MN, the central computer sends this information to the corresponding AN referred by the list. This selected AN is in charge of delivering the data to the MN.

On the other hand, when a MN wants to transmit information to the central computer, it sends a packet containing its long address and the corresponding information to a certain AN. This anchor routes the packet to the central computer by executing the described routing algorithm. The configuration list is updated every time a packet, related to a MN, arrives at the central computer.

III. Protocol Description

In this section, we firstly describe the principal operation of HCP. Afterwards, we give also details about the way, in which the ANs communicate with each other and the central computer. Additionally, a brief description about the implemented synchronization process is given. For further information, please refer to [1].

A. *Highly Configurable Protocol*

HCP was created to allow the MNs take different configurations, according to the context. Each MN's configuration presents advantages or drawbacks in terms of energy consumption, generated traffic in the network and position accuracy. Thus, by changing the operation mode, the MNs can balance the trade-off between performance and used resources, according to the demand. For example, if a MN works in a certain operation mode that provides high position accuracy, and then, it detects that its battery level is critic, the MN can select another operation mode that allows it to extend the battery's lifetime, but maybe, at the expense of reducing the position accuracy.

In our previous work [1], four different MN's configurations were presented. However, in this paper, we focus only on the performance evaluation of the configurations "mode 1" and "mode 3", because they are significantly different from one another and the remaining configurations can be considered as variations of these operation modes.

In the configuration mode 1 the ANs broadcast packets and the MN takes RSSI samples. On the other hand, in the configuration mode 3, the MN transmits packets and the ANs take RSSI samples.

In HCP, the time domain is divided into periods, and these into phases (see Fig. 3). The duration of the period and phases are known by all nodes in the network. By using a synchronization algorithm, the nodes know when a new period starts.

As the Fig. 3 indicates, a period contains three phases called SYNC. During these phases the ANs broadcast packets with synchronization information and their coordinates. These packets can be used by the MN to take RSSI measurements.

In each period there is also one REPORT phase, where the MNs can inform about their measurements and receive information from the ANs.

The VIP phase is reserved for MNs that require consuming as little energy as possible. Here, the MNs can broadcast packets and the ANs take RSSI samples. Finally, the communication between the ANs and the central computer take place in the COM_SINK phase.

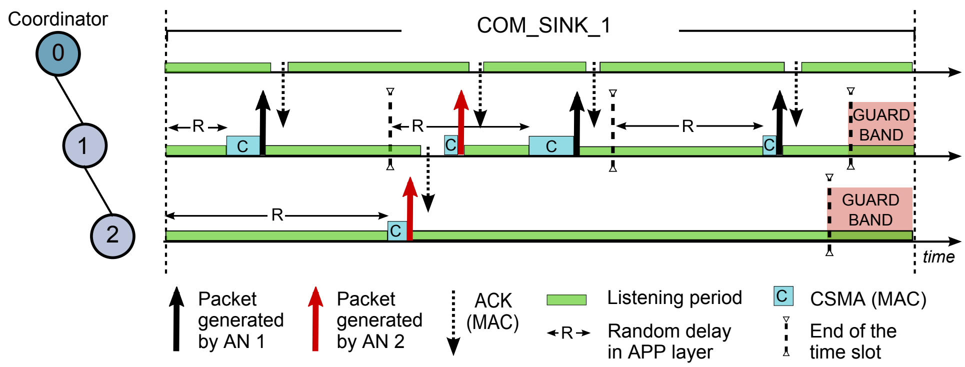

Fig. 4. Operation of the ANs during COM_SINK_1.

The selected operation mode defines the specific schedule and the tasks carried out by the MN in the phases.

In mode 1, the MN listens to packets during one or more SYNC phases to take a certain number of RSSI samples. Between SYNC phases the MN sleeps to save energy.

In REPORT phase, the MN waits a random delay and wakes up to send a packet (containing the measurements) to a specific AN. Normally, the MN selects the AN with the strongest RSSI. Once the MN receives the MAC ACK from the selected AN, it goes into sleep mode waiting for the next active period. The random delay is to distribute the transmissions across the phase. This minimizes collisions and avoids large backoff periods (CSMA) when several MNs want to transmit in the same REPORT phase [13].

Afterwards, the selected AN has two options: either it can use the RSSI measurements to estimate the position of the MN, or it can send this information to the computer during the COM_SINK_1 phase for centralized calculation. In this work, we have investigated the second option.

In mode 3, the MNs sleep during large periods saving energy and wake up (randomly over time) to broadcast a certain number of short packets during the VIP phase and then, sleep again. Only when the MN needs to resynchronize to the network, it listens during a short time in phase SYNC to receive just one packet with synchronization information.

The ANs, which have taken RSSI samples from the broadcasts, send this information to the central computer during the COM_SINK_1 phase. Finally, the central computer estimates the position of the MN.

B. COM_SINK phase

One of the most important tasks, in the centralized localization process, is the sending of the gathered information to the central computer for the position estimation. In this section, we explain the COM_SINK phase in detail, where the multi-hop communication between the ANs is carried out to deliver data to the central computer.

The ANs are principally in charge of two tasks in the COM_SINK phase. They generate packets with information about the MNs and route the packets that arrive from their parents or children.

In our implementation we have created two COM_SINK phases. In the first one (COM_SINK_1), the uplink communication to the central computer is carried out. The COM_SINK_2 phase is reserved for the downlink communication, in case the central computer has to send some information to the ANs.

In COM_SINK_1 phase, when the ANs have to generate packets related to the MNs, they define as many time slots as packets to be generated. All time slots have the same duration. Within each time slot, the AN waits for a random delay in the application layer and transmits one of its transmission requests to the lower layer. If the packet is transmitted successfully and the corresponding MAC ACK is received, the application layer receives a confirmation primitive and continues with its planned tasks.

In case the AN suddenly receives a packet from its child AN and this has to be routed, the routing algorithm decides the next hop and immediately sends a transmission request to the MAC layer, which executes the final tasks for the effective transmission to the next AN.

Fig. 4 shows an example of the operation of two ANs and the coordinator during COM_SINK_1 phase. They are located in the same branch of the tree topology. In the case, that the coordinator with address 0 does not need to inform about any MN, the AN with address 1 has RSSI information of three MNs, thus, it defines three time slots in the COM_SINK phase for the transmissions of its packets. Otherwise, the AN with address 2 has RSSI information from one MN, therefore it create just one slot to generate its packet. In this example, it can be seen how the packet which is received from AN 2, is immediately routed by AN 1 without waiting a random delay in the application layer.

A problem occurs when an AN receives a packet and there is no more time in the phase to route it to the destination. In order to avoid this, a band guard is included at the end of the COM_SINK phase. In this guard band, the ANs cannot generate transmission requests of their own packets. They are only able to route packets from other ANs. The duration of the guard bands can be defined depending on the hierarchical level of the AN. Thus, the ANs of deeper levels will have longer guard bands.

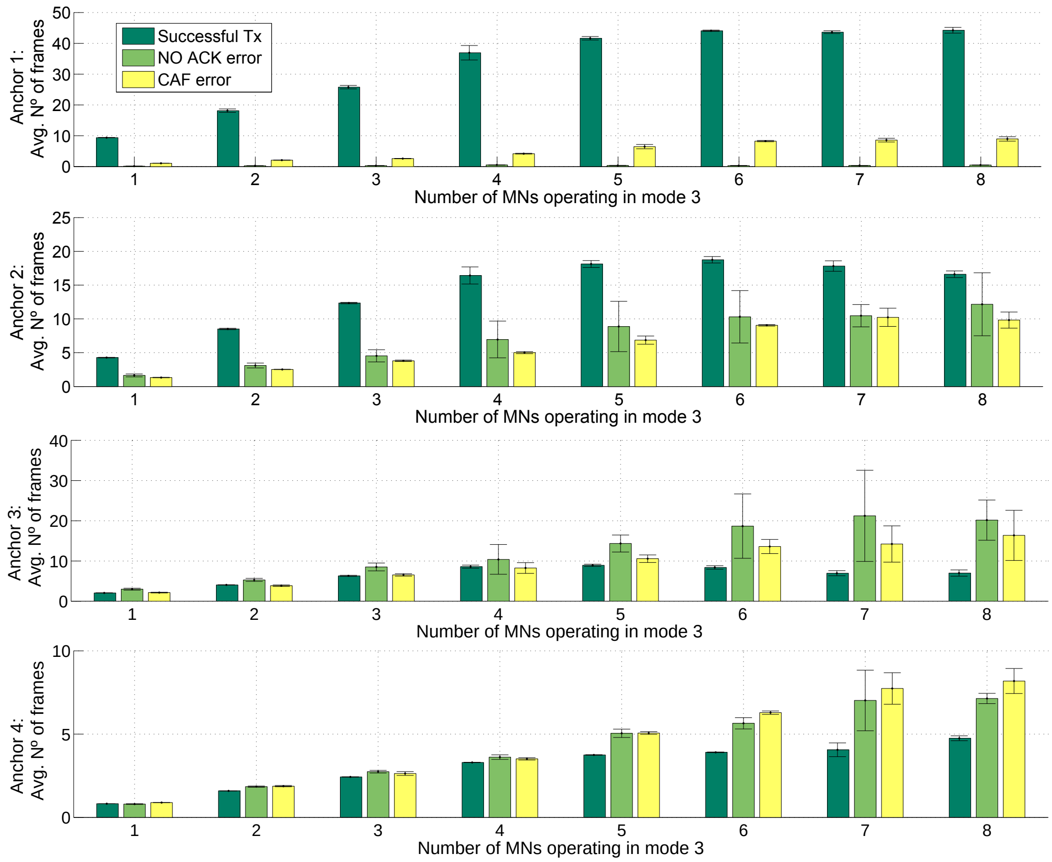

Fig. 5. MAC errors in the ANs when the MNs operated in mode 3 [10].

C. Synchronization algorithm

The coordinator determines the instant of time in which the periods start. This information is propagated to the deeper levels of the tree topology to synchronize the network.

When a new AN has joined to the network (after finishing its association process), it goes into listening mode waiting for a synchronization packet from its parent node. The synchronization packets, which are transmitted in the SYNC phase, contain the remaining time to the next phase. Thus, the new AN synchronizes to the network by using this information and the duration of the next phase. As above mentioned, the duration of the phases are configuration values known by all nodes in the network. In [1], a detailed explanation of the synchronization protocol of HCP is given.

After the AN was synchronized to the network, it is able to broadcast packets during the SYNC phases. The synchronization of the ANs is updated when they listen to new synchronization packets from ANs located at higher hierarchical levels.

In a similar way, when a new MN is switched on, it is in listening mode to wait for a synchronization packet from any AN. Once the MN is synchronized, it is able to operate within the network according to its operation mode by default.

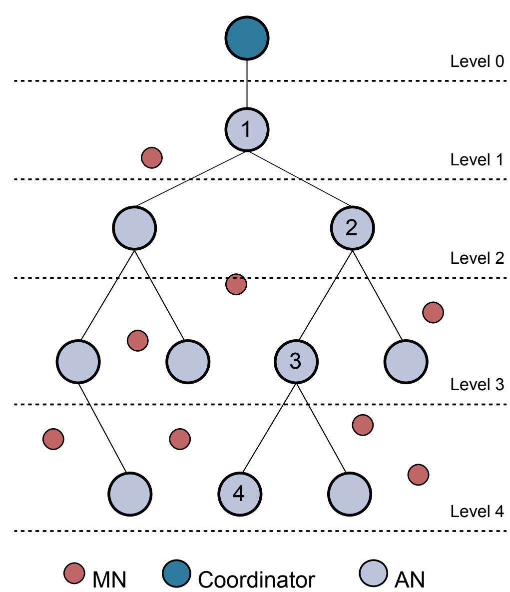

Fig. 6. Reference scenario.

IV. Evaluation

In this section, we present and discuss the results related to the total number of packets arrived at the central computer and the MAC errors (CAFs, NO_ACK) registered at the ANs during the COM_SINK_1 phase.

For the performance evaluation presented here, two different tests were performed. In the first one the MNs operated in mode 3, while in the second test, the MNs operated in mode 1.

At the end of the section, we compare both operation modes in terms of the amount of localization information refereed to a specific MN in the network.

A. Reference Scenario and Methodology

Fig. 6 shows the reference network used in our analysis. It consists of one coordinator, 10 ANs located in 4 different hierarchical levels and up to 8 end devices, which operated in mode 1 or in mode 3 depending on the evaluation case.

In this test scenario, the MNs and the ANs are static. Furthermore, each node is in the transmission range of all other nodes.

In order to investigate the internal operation of the ANs, different counters are included in the software layers, e.g. there is a counter that registers the number of CAFs occurred during the COM_SINK_1 phase. These counters are reset at the beginning of each period.

After the execution of a period, a "debugging phase" is carried out, where the central computer sequentially asks each AN of the network for receiving information about its internal counters. If the computer does not receive the corresponding ANs answer, it tries again. Once the computer receives the ANs answer correctly, it continues with the next AN. The duration of the debugging phase is 10 seconds.

The central computer was programmed to classify and store the received information. Observe that the debugging phase operates only during our evaluation, but it does not form part of the original HCP. During the debugging phase the MNs are sleeping.

The durations of the phases of HCP were configured as following: 60ms for SYNC, 400ms for REPORT, 300 ms for VIP, 1000ms for COM_SINK_1 (uplink) and 500ms for COM_SINK_2 (downlink). In the COM_SINK phases, a constant guard band of 100ms is considered.

A maximum of 3 and 4 retransmissions are allowed in the MAC and NWK layers, respectively. The default configuration of the MAC layer proposed in the standard is used in the tests.

The measurements of more than 300 independent periods were stored to obtain average values and confidence intervals (confidence level: 95%).

B. Mode 3

As above mentioned, the MNs operated in mode 3 in our first test. Fig. 7 shows the average number of packets received by the central computer (per period) during the COM_SINK_1 with different number of transmitting MNs.

The curve *Ideal* depicts the case without the occurrence of errors. Thus, for example, if there are 6 transmitting MNs, the computer should receive 60 packets with RSSI information, because all ANs are in the transmission range of the MNs, and each AN generates a packet per MN.

In order to receive 60 packets at the central computer, 60 packets have to be generated and 114 packets have to be routed by the ANs in the multi-hop communication. Thus, the transmissions of 174 packets and 174 ACKs (without considering retransmissions) have to be distributed during the COM_SINK_1 phase. The approximate duration of the packets and ACKs are 1ms and 0.4 ms, respectively. Therefore, in this ideal case, the channel should be occupied around 243 ms during this phase.

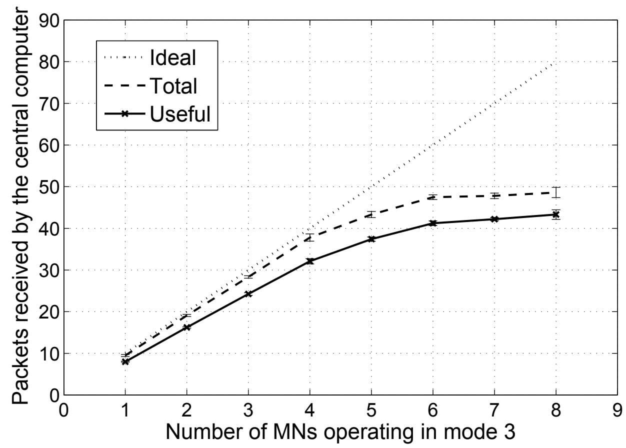

Fig. 7. Average number of packets received at the central computer per period, when the MNs operated in Mode 3 [11] .

Our results show that the reality is far from the ideal case. The central computer does not receive all generated packets. The reasons will be given later in this section. In Fig. 7, the curve *Total* depicts the average number of packets that arrived at the central computer. The curve *Useful* represents the total number of packets without considering the duplicated packets.

Duplicated packets can occur when an AN receives a packet and transmits its corresponding ACK, but this ACK is not correctly received e.g. due to a collision with other transmission. In this case, the transmitter tries a retransmission. Thus, it is possible that an AN receives the same packet for the second time. The implementation of a filter at the ANs that identifies and discards duplicated packets could improve the network performance. This task is planned for a future work.

Fig. 7 shows that after 5 transmitting MNs, there is no significant increase in the total number of received packets by the central computer. It is important to note, that the number of packets transmitted by the ANs strongly depends on the CSMA delay. Previous measurements show that, in some cases, the CSMA delay can be longer than 9 ms [14]. During this random time, our reference nodes are not able to listen to another packet. Thus, the transmission of 10 packets could mean that the AN is occupied during more than 100ms. Furthermore, if ACKs and retransmissions are allowed, then the ANs can require more time to send a packet successfully. These factors reduce the available listening time of the ANs in the COM_SINK phase.

Assume that an AN has a child node and they want to

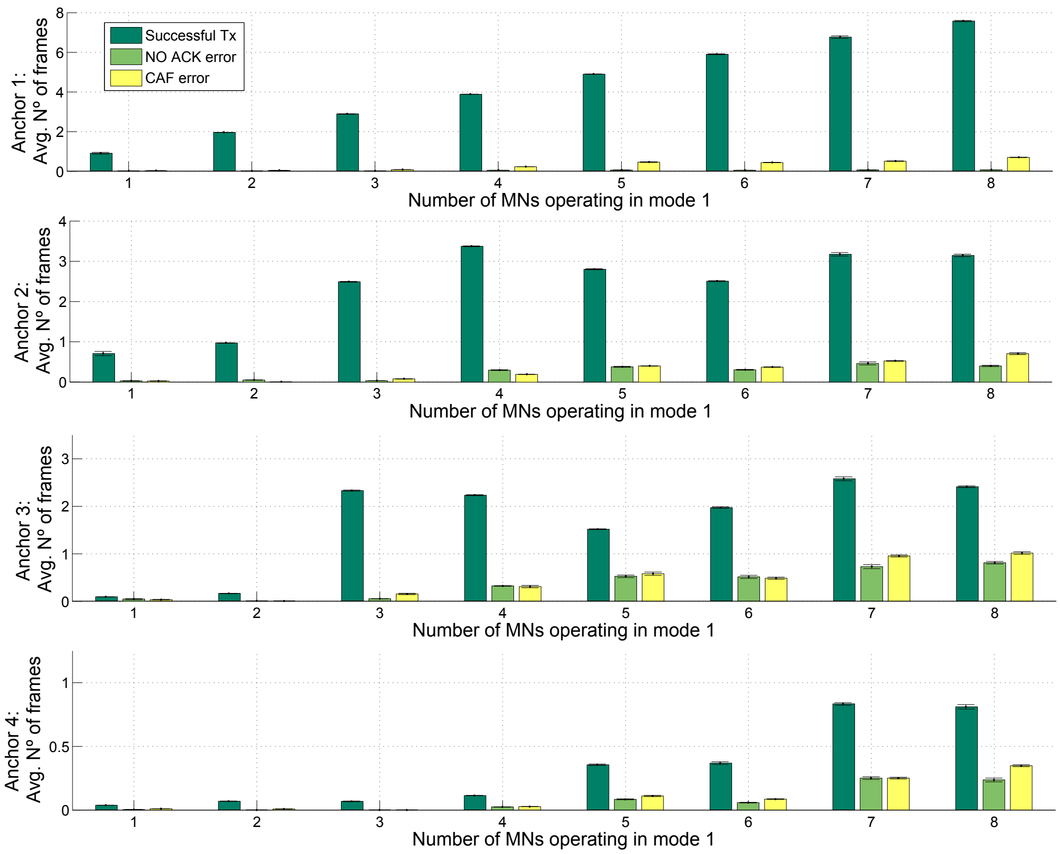

Fig. 8. MAC errors in the ANs when the MNs operated in mode 1.

transmit a packet at the same time. It is possible that a child finds the channel free when the parent node is in CSMA waiting for a random delay. In this case, the child node transmits a packet to its parent node, which cannot be received because the parent executes CSMA. Due to the fact that the child does not receive an ACK, it retransmits the packet. In turn, the transmissions of the child node can mean that the parent node cannot find the channel free generating a new backoff delay and extending the duration of its CSMA mechanism. This process can be repeated many times leading to the generation of CAF errors at the parent node and NO_ACK errors at the child node.

Furthermore, if two nodes perform CCA exactly at the same time, then, they can detect the channel free and transmit their packets simultaneously leading to a collision. It is also possible that a node perform CCA during a transition time (e.g. between the reception of a packet and the generation of its ACK) of another node. Thus, the transmission of a node can collide with an ACK from other node.

Another problem occurs when the inter-transmission time of two consecutive packets is very short. Assume that the child node achieves to transmit a packet to its parent node, then, in the next step the parent node has to route this packet. Errors can occur if the child node tries to transmit a new packet while the parent node is trying to route the previous packet. This problem affects the parent as much as the child node. It could be minimized if the child waits for a time after a new transmission. This minimum inter-transmission time could be estimated by using the CSMA delay information of previous periods.

Fig. 5 depicts the average number of MAC errors registered in four ANs of the network (short address: 1, 2, 3 and 4). They are located in different hierarchical levels in the same branch of the tree topology (see Fig. 6). As expected, the number of MAC errors increases with increasing number of transmitting MNs. After 5 MNs, the network is saturated and therefore, from this point the same amount of transmitted packets and MAC errors are shown.

The AN 1 is directly connected to the coordinator. During the COM_SINK_1, the coordinator receives the information from AN 1 and sends it to the central computer via USB. While the microcontroller transfers the data to the central computer, the transceiver is available to receive another packet. In this way, most of the time the transceiver of the coordinator is available for its child to receive data and answer with ACKs. Therefore, Fig. 5 shows very few NO_ACK errors in AN 1. The MAC errors registered at AN 1 are mainly because of the channel congestion (CAFs).

As described, there is a significant relationship between the errors of parents and children. In fact, the long times required by the parent node in its transmissions, negatively impact on the transmissions opportunities of its child node increasing the occurrence of errors. This can be demonstrated by analyzing the results of AN 2 and 3, where AN 3 presents more errors than AN 2. In general, the ANs which are located in deeper levels of the tree topology had more problems to send its data through the network to the central computer.

The AN 4 does not have children; therefore, this node has to transmit fewer packets compared to the other ANs. These packets can be distributed throughout the entire COM_SINK_1 phase allowing long separations between them and thus, avoiding the problems caused by short inter-transmission times. Despite that, many errors were detected in our tests due to the strong congestion of its parent node (AN 3) and the occupancy of the channel. Indeed, in case of 8 MNs, the AN 4 registered 15 MAC errors and the successful transmission of 5 packets (on average, per period).

C. Mode 1

In the ideal case, if there are 8 active MNs operating in mode 1, the central computer should receive 8 packets per period. Fig. 9 refers to this test. The legends used in this figure are the same as in Fig. 7. Note, that the total number of received packets and the ideal case are similar. However, a small increase of packet losses is detected, when the traffic increases in the network.

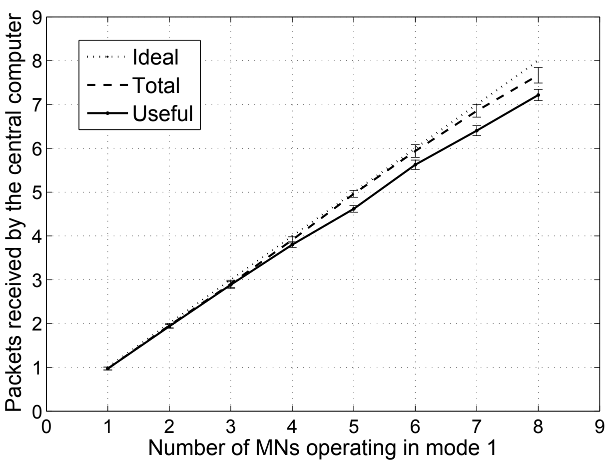

Fig. 9. Average number of packets received at the central computer per period, when the MNs operated in Mode 1 [11].

Fig. 8 shows the number of MAC errors in our four reference ANs, while the MNs operated in mode 1. The figure indicates that the ratio between the transmitted frames and the total errors registered at the ANs is more favorable than in the case when the MNs operated in mode 3. For instance, considering the case with 5 transmitting MNs, the number of transmitted frames by AN 2 in the first test was almost the same as the sum of registered MAC errors (see Fig. 5). In the second test, when the MNs operated in mode 1, the number of registered errors was around 30% of the transmitted frames.

Although the network was not completely saturated in this second test, errors occurred due to the already mentioned problems (short inter-transmission time, collisions, etc). Note, that as in the above case, the CAF errors are more frequent than the NO_ACK errors in the AN 1, while the occurrence of these kind of errors are similar for the other three ANs.

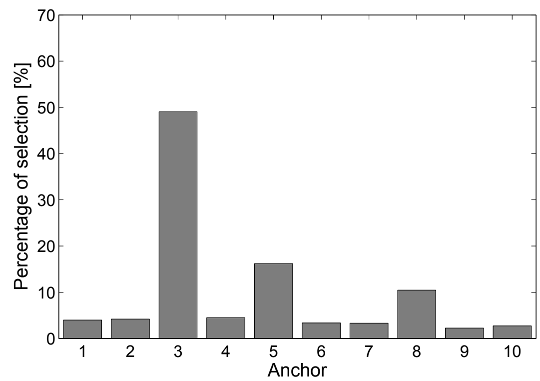

Fig. 10. Preference of the reference MN in the selection of an AN to transmit the data to the central computer (mode 1) [11].

As all nodes are neighbors and static in our test scenario, the MNs do not select always the same AN. It is due to the random fluctuations of the RSSI measurements [14]. Fig.10 shows, in percentage, which AN was preferred by a specific MN (hereinafter referred to as reference MN) for reporting its measurements.

During the test, the reference MN was situated very close to AN 3 and therefore, its related RSSI was often the strongest one. However, the results also show how this MN selected other ANs when, due to the above mentioned RSSI fluctuations, the RSSI from other AN (e.g. 5 or 8) was stronger than the registered one from AN 3.

D. Comparison Mode 1 - Mode 3

The big advantage of mode 3 is the low energy consumption at the MNs [1]. However, this mode can generate the transmission of multiple packets in a limited sensor network reducing the scalability.

The loss of packets with localization information impacts the position accuracy in a different way, depending on the used localization algorithm. We have investigated how many packets with RSSI information of the reference MN, are received per period at the central computer. Fig.11 shows the resulting histograms when there are 2, 4, 6 and 8 transmitting MNs (mode 3). In the ideal case the central computer should receive 10 packets related to this MN in all periods, because there are 10 ANs that listen to the MN's broadcasts. Thus, the histograms should show a bar (relative frequency equal to 1), indicating that central computer received 10 packets in each period.

In the real case, there are packet losses. We have detected that when there are fewer than 5 transmitting MNs, the network seems to be not saturated and the resulting histograms are similar (e.g. compare Fig. 11 a) and b)). After this point, the different histograms show how the mean value is shifted downward with increasing number of transmitting MNs. It can be seen, that there is small probability of receiving 10 packets related to the reference MN, when this MN operates together with other 7 MNs (Fig. d)).

These last measurements suggest that the number of useful packets (related to the same MN) that arrive at the central computer could be modeled by using a discrete normal distribution, whose mean value depends on the number of transmitting MNs (mode 3). Naturally, this function should be truncated by the maximum number of active ANs (10).

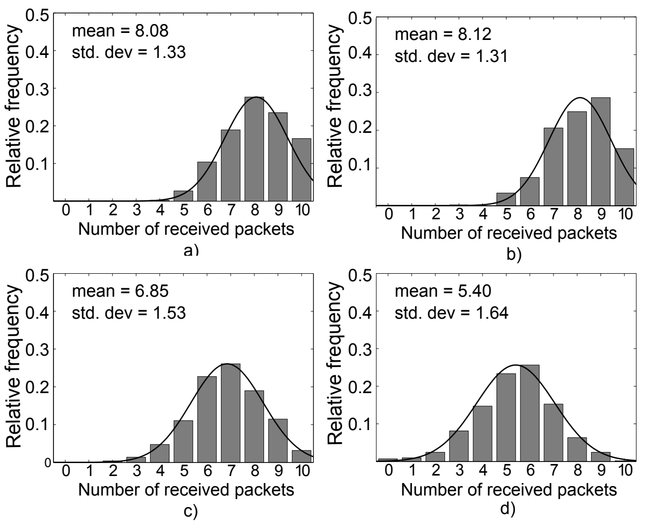

Fig. 11. The histograms illustrate how many packets related to the reference MN arrived at the central computer in each period, when there were a) 2 , b) 4, c) 6 and d) 8 transmitting MNs (mode 3) [11].

It is well known that the performance of the localization algorithms can be improved if many measurements and ANs are used for the estimation [15]. Furthermore, there are localization algorithms that need a minimum number of measurements to operate correctly. For instance, Multilateration needs the distance measurements from at least 3 ANs to calculate the MN's position in a 2D scenario [16]. Thus, it is important to investigate if the central computer receives the necessary information for the estimation.

In mode 1, the central computer should receive one packet (containing the RSSI measurements of 10 ANs) per MN in each period. In case this packet is lost, there is no information to estimate the position in this period. The Fig. 12 depicts the relative frequency of receiving such packet at the central computer when the MNs operated in mode 1. It can be shown that in the most cases the packet arrived at the central computer (because the network is not saturated). Note, that with 8 transmitting MNs, the packet arrived in the 90% of the investigated periods.

The situation changes when the nodes operate in mode 3. Fig.12 also shows the relative frequency of receiving packets (related to the reference MN) from "at least" 3, 6, 8 and

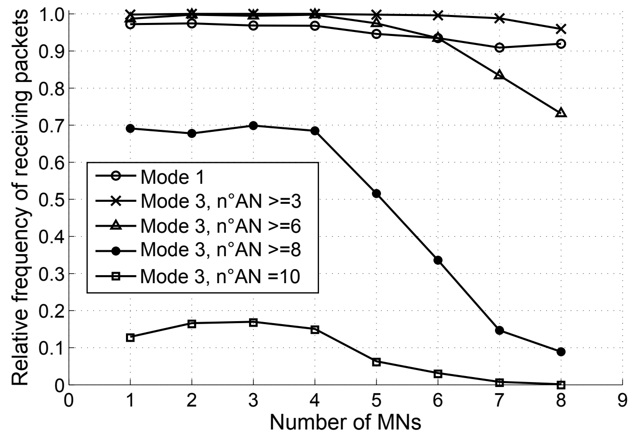

Fig. 12. Relative frequency, in receiving localization information about the reference MN, from a certain amount of ANs. Comparison between mode 1 and mode 3 [11].

10 ANs, with increasing number of transmitting MNs. Both Fig.11 and Fig.12 indicate that it is very difficult that the computer receives all packets generated by the 10 ANs. Even with one MN operating in mode 3, the central computer received 10 packets in less than 20% of the cases. On the other hand, the results also suggest that it is very probable that the computer obtains the RSSI information from at least 3 ANs. In the worst case with 8 MNs, a relative frequency of around 0.95 was registered (see Fig.12).

V. Conclusion

In this work, two MN's operation modes of the protocol HCP [1] were deeply investigated in a 802.15.4 sensor network.

In the first one (mode 3), the MNs broadcast packets and the ANs take signal strength samples. Then, the ANs send this data via multi-hop communication to a central computer for the position estimation.

In the second configuration (mode 1), the anchors broadcast packets and the MNs take signal strength measurements. Then, each MN sends a report with its measurements to a certain AN, which sends this information to the central computer. Note, that HCP provides a common framework that supports different MN's configurations at the same time.

We have investigated how many packets with location information arrived at the central computer with increasing number of MNs and for different configurations of the network. This information can be used, in a next step, to investigate the corresponding degradation of the position accuracy given a certain localization algorithm. It was also analyzed, how the hierarchical level of the anchors (tree topology) impacts on the communication.

Multiple packet losses were registered in our reference scenario, when more than 5 MNs operating in mode 3. The advantage of this mode is the low MN's power consumption, because the MN sleeps during long periods. Contrarily, in mode 1, the MN has to be in listening period for taking RSSI samples and this consumes more energy in comparison with mode 3 [13]. Observe that the traffic generated by the MNs in mode 1 is lower than in mode 3, increasing the scalability of the network.

The principal causes of the detected problems (packet losses, duplicated packets and congestion) were: the transmission of two consecutive packets to the parent node, the collisions between packets because two nodes detected the channel free at the same time during CSMA, the reduction of the available listening period due to long CSMA delays and the ACK losses in the MAC level.

In a future work, it is planned to minimize these errors, e.g. by including different mechanisms to filter duplicated packets, change the parent node dynamically according to the presented congestion and avoid short inter-transmission time.

VI. Acknowledgment

The authors would like to thank Sebastian Tromer for his help during the implementation. Special thanks are due to Jorge Pérez Hidalgo and Víctor Casas Melo for their feedbacks and suggestions. Last, but not least, the support received from the Gesellschaft von Freunden und Förderern der TU Dresden is gratefully acknowledged.

References

[1] J.J. Robles, S. Tromer, J. Pérez Hidalgo and R. Lehnert, *A High Configurable Protocol for Indoor Localization Systems*, In Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2011, Guimarães, Portugal, Sep 2011.

[2] H. Liu, H. Darabi, P. Banerjee and J. Liu,*Survey of wireless indoor positioning techniques and systems*, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 37, No. 6, 1067-1080, 2007.

[3] IEEE 802.15 WPAN Task Group 4, *IEEE standard 802.15.4-2006*, available at (08.2012) http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf

[4] Dresden Elektronik, *Datasheet sensor node RCB230*, available at: (08.2012) http://www.dresden-elektronik.de/shop/prod68.html.

[5] Atmel corporation, *Datasheet microcontroller Atmega1281*, available at: (08.2012) http://www.atmel.com/Images/doc2549.pdf

[6] Atmel corporation, *802.15.4 Transceiver AT86RF230*, available at: (08.2012) http://www.atmel.com/Images/doc5131.pdf

[7] Atmel corporation, *Application Note: Atmel AVR2025: IEEE 802.15.4 MAC Software Package - User Guide*, available at (08.2012) http://www.atmel.com/Images/doc8412.pdf

[8] Zigbee Alliance, *ZigBee-2006 specification, ZigBee document 064112, 2006.*

[9] S. Tromer, *Implementation of an Energy Efficient Indoor Localization Algorithm*, Student thesis, Supervisor: J. J. Robles, Chair for Telecommunications, Technische Universität Dresden, Feb 2010.

[10] L. de la Cuesta Ibañez, *Performance Investigation of an Indoor Sensor Network*, Diploma thesis, Supervisor: J. J. Robles, Chair for Telecommunications, Technische Universität Dresden, Nov 2011.

[11] E. Gago Muñoz , *Implementation of a Highly Configurable Protocol for Indoor Localization*, Diploma thesis, Supervisor: J. J. Robles, Chair for Telecommunications, Technische Universität Dresden, Nov 2011.

[12] M. Pan,H. Fang, Y. Liu and Y. Tseng, *Address Assignment and Routing Schemes for ZigBee-Based Long-Thin Wireless Sensor Networks*, Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE , pp.173-177, 11-14 May 2008.

[13] J.J. Robles, S. Tromer, M. Quiroga and R. Lehnert, *Enabling Low-power Localization for Mobile Sensor Nodes*, In Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2010, Zürich, Switzerland, Sep 2010.

[14] J.J. Robles, *Considerations in the Design of Indoor Localization Systems for Wireless Sensor Networks*, 17th EUNICE International Workshop, Dresden, Sep 2011.

[15] P. Moravek, D. Komosny, M. Simek and J. Muller, *Multilateration and Flip Ambiguity Mitigation in Ad-hoc Networks*, Przegl Ad Elektrotechniczny, ISSN 0033-2097, R. 88 NR 5b/2012.

[16] A. Savvides, H. Park and M. Srivastava, *The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems*, First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA, 2002, pp. 112-121.

[17] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.

[18] G. Mao and B. Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks*, Information Science Reference, 2009.