Paper

Developments on an IEEE 802.15.4-based wireless sensor network

Bart Scheers, Wim Mees, and Ben Lauwens

Abstract— In this paper a summary is given of the ongoing research at the Belgian Royal Military Academy in the field of mobile ad hoc networks in general and wireless sensor networks (WSNs) in particular. In this study, all wireless sensor networks are based on the physical and the medium access layer of the IEEE 802.15.4 low rate wireless personal area networks standard. The paper gives a short overview of the IEEE 802.15.4 standard in the beaconless mode together with a description of the sensor nodes and the software used throughout this work. The paper also reports on the development of a packet sniffer for IEEE 802.15.4 integrated in wireshark. This packet sniffer turns out to be indispensable for debugging purposes. In view of future applications on the wireless network, we made a theoretical study of the effective data capacity and compared this with measurements performed on a real sensor network. The differences between measurements and theory are explained. In case of geographically meaningful sensor data, it is important to have a knowledge of the relative position of each node. In the last part of the paper we present some experimental results of positioning based on the received signal strength indicators (RSSI). As one could expect, the accuracy of such a method is poor, even in a well controlled environment. But the method has some potential.

Keywords— wireless sensor networks, IEEE 802.15.4, effective data capacity, positioning.

1. Introduction

Wireless ad hoc network is a generic term grouping different networks, which are self organizing, meaning that there is neither a centralized administration nor a fixed network infrastructure and that the communication links are wireless. Different types of wireless ad hoc networks include mobile ad hoc networks (MANETS), wireless sensor networks (WSNs), smart dust, etc. A wireless sensor network is an ad hoc network consisting of spatially distributed autonomous sensor nodes, i.e., nodes equipped with a radio transceiver, a microcontroller, an energy source (usually a battery) and a sensor, to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations (see Fig. 1).

Wireless sensor network differ from classical ad hoc networks in several ways, e.g., the number of nodes is larger and the spatial distribution of the nodes is more dense, the nodes are normally static (however, this is not always the case), the energy of the nodes is limited, the amount of data transiting through the network is limited and in most cases the data is converging to one single server node, collecting and processing the data. All these factors have their influence on the choice of the technology and routing protocol used in this type of ad hoc networks.

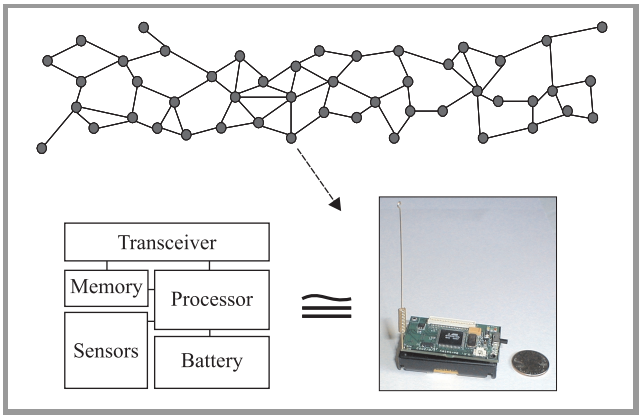

Fig. 1. Wireless sensor network.

The paper will be organized as follows. In Section 2 we will give some background on the IEEE 802.15.4 PHY and MAC layer, the sensor nodes and the software that is used throughout this research. In Section 3 we will report on the development of a packet sniffer for an IEEE 802.15.4-based wireless sensor network. In Section 4 we will discuss the theoretical effective data capacity and compare this with measurements conducted on a real sensor network. In the last section we will describe how we can estimate the relative position of a sensor node in the network, based on the received signal strength indicators (RSSI) from beacon nodes with a priori known position. We will show the result of measurements conducted on a real sensor network, deployed on a football field, and discuss the accuracy of such a method.

2. Background

2.1. The IEEE 802.15.4 standard

The IEEE 802.15.4 is a recent standard, approved in 2003, describing the physical (PHY) and medium access control (MAC) layers for low rate wireless personal area networks (LR-PAN) [1]. IEEE 802.15.4 is expected to be deployed on massive numbers of wireless devices, which are usually inexpensive, long-life battery powered and of

low computation capabilities. As such, the standard is also ideal for WSN. At the physical layer the standard provides for the use of 3 frequency bands. The most popular one being the 2.4 GHz industrial, scientific and medical (ISM) frequency band. In this frequency band, 16 channels are available, each with a data throughput of 250 kbit/s on the physical layer. On the MAC layer, the IEEE 802.15.4 standard supports different modes of operation: beacon-enabled or beaconless network mode, with or without a PAN coordinator, in a star or in a peer-to-peer topology. Almost all combinations of these 3 couples are possible.

In the scope of this research, we only use the beaconless network mode, without a PAN coordinator in a peer-to-peer topology. Note that this mode of operation allows multiple hops to route messages from any device to any other device. These routing functions can be added at the network layer, but are not part of the standard. As we only use the beaconless network mode without a coordinator we will limit the explanation of the medium access protocol to this particular mode. In a beaconless network, the medium access is, just as in WIFI, based on un-slotted carrier sense multiple access – collision avoidance (CSMA-CA). However, unlike the IEEE 802.11 standard, IEEE 802.15.4 omits the request/clear to send (RTS/CTS) exchange; hence the hidden node problem will be an issue. The omission of the RTS/CTS frames is justified by the limited size of the MAC data packet unit, with is fixed to a maximum of 127 bytes in the standard.

Figure 2 shows a communication between two network devices in a beaconless mode. Source device A first performs a clear channel assessment (CCA) is used to verify whether the medium is free or not. If the channel is free, the source device will send out the data frame and wait for an acknowledge frame (optional). All other nodes, overhearing this communication, will defer their transmission. In case of an occupied channel, an exponential backoff mechanism is used.

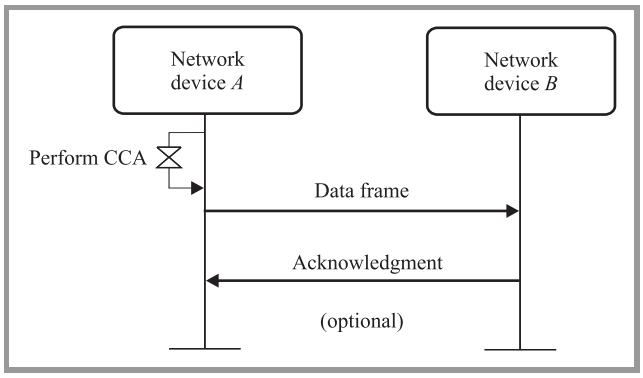

Fig. 2. Communication between two devices.

The MAC layer of the device trying to get access to the medium will delay its transmission for a random number of complete backoff periods in the range 0 to $2^{BE} - 1$. *BE* is the backoff exponent and a unit backoff period equals 320 μs in the 2.4 GHz band. If, after this delay, the channel is assessed to be busy again, the MAC layer will increment *BE* by one until *BE* reaches the value of 5 (maximum value for *BE*). The initial value of *BE* can be set by the user. Note that if *BE* is initialized to 0, collision avoidance will be disabled during the first attempt to access the medium.

Each device (transmitter) is identified by a unique 64 bit hardware address, called the extended address, comparable with an Ethernet MAC address. The standard however allows the allocation of a 16 bit short address, which considerably reduces the addressing fields in the MAC frame. More details on the structure of the data frame will be given in Section 4.

2.2. The sensor nodes

The hardware platform that is used as building block for the WSN is the Tmote™ Sky platform from Moteiv [2] (see Fig. 3). The Tmote Sky platform is a wireless sensor node based on a TI MSP430 microcontroller with an IEEE 802.15.4-compatible radio chip CC2420 from chipcon [3], with an on-board antenna. The Tmote Sky platform offers a number of integrated peripherals including a 12-bit ADC and DAC and a number of integrated sensors like a temperature sensor, 2 light sensors and a humidity sensor.

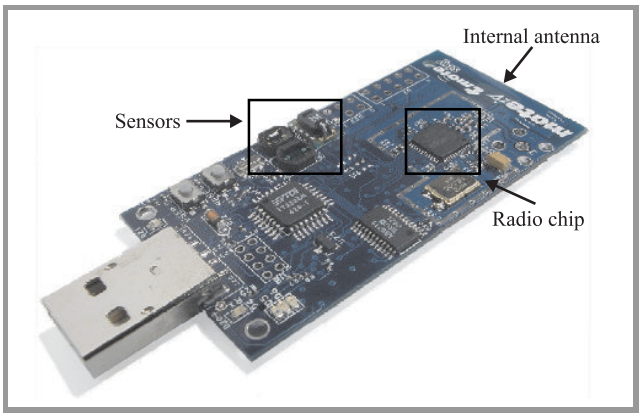

Fig. 3. Tmote™ Sky platform from Moteiv.

The microcontroller is programmed through the onboard universal serial bus (USB) connector, which makes it easy to use; no additional development kit for the microcontroller is needed. The USB can also be used as a serial port to communicate with a host computer.

2.3. The real time operating system and communication stack

Throughout all the projects, Contiki is used as real time operating system on the Tmote Sky sensor nodes.

Contiki is an open source multi-tasking operating system for networked systems. It is designed for embedded systems with small amounts of memory. A typical Contiki configuration is 2 kbytes of RAM and 40 kbytes of ROM. Contiki consists of an event-driven kernel on top of which application programs can be dynamically loaded and unloaded at runtime. The main reason why Contiki was

chosen as real time operating system (RTOS) is that it is written in standard C, which makes it easy to understand and to modify.

As almost all applications on military networks are IP based, we opted to use a TCP/IP stack on top of the IEEE 802.15.4 devices and not the usual ZigBee stack.

Contiki contains a small request for comments (RFC)-compliant TCP/IP stack that makes it possible to communicate over an IP enabled network. Contiki also contains a RFC-compliant ad hoc on-demand distance vector (AODV) routing protocol. AODV is a reactive routing protocol for ad hoc networks. In a reactive routing protocol, routes are only created when desired by the source nodes. When a node requires a route to a destination, it initiates a route discovery process within the network. This process completes once a route is found or all possible route permutations are examined. The route is maintained only if there are data packets periodically travelling from the source to the destination along that path. This protocol is what is called "source initiated".

3. Development of a packet sniffer

Doing research on IEEE 802.15.4 enabled WSN, it is indispensable to have a good packet sniffer for debugging purposes.

At the time this research started, the only available packet sniffer was the chipcon packet sniffer for IEEE 802.15.4 which comes with the CC2420 evaluation board. The evaluation board is connected through the PC with a USB cable. The board is able to queue up to 248 packets for USB transfer, allowing short periods of high workload for the PC. A large amount of packets can be stored on the computer in a trace file using a specific format.

Unfortunately the CC2420 packet sniffer only analyses the PHY and MAC layer and not the IP data transported in the MAC frame. We therefore developed a packet sniffer that can be integrated in wireshark. Wireshark, formerly known as Ethereal is a free software protocol analyzer.

As the IEEE 802.15.4 standard was not yet supported by wireshark, we first had to write a plug-in, in order to be able to correctly decode the IEEE 802.15.4 frames. Wireshark uses dissectors, identified by a DLT_number, to decode a specific layer or protocol, hence a new DLT_number had to be requested for this new link-layer protocol to the developers of wireshark. The value 191 (0xBF) was attributed by them. Based on this DLT_number a dissector was written to decode the IEEE 802.15.4 data and acknowledge frames. Once decoded, the LL payload is then passed to the next dissector (IP in our case).

The files that can be imported and decoded by wireshark must be libpcap compatible. To obtain these pcap files, we worked out two solutions. The first solution is based on the earlier presented CC2420 packet sniffer. A software was written to transform the trace file from the CC2420 sniffer into a libpcap compatible file format which could then be imported in wireshark. The second solution is based on the Tmote Sky sensor node. The software, downloaded on the Sky node, puts the IEEE 802.15.4 radio in promiscuous mode and does a continuous copy of the frames, received on the air interface, to the USB serial interface. A PC, connected to the node, runs a program that reads the USB interface and writes the content of the PHY payload immediately to a libpcap compatible file.

In the first solution, the representation of the captured frames in wireshark is done in three steps; first the capturing by the chipcon sniffer, then the conversion to a pcap file. Once this is done the pcap file can be imported and decoded by wireshark. In the second solution, the analysis is done in two steps as the received frames are directly written to a libpcap compatible file. The development of the latter solution is still ongoing. For the moment, the timestamp of the arriving frames is given by the PC. However, due to the limited data rate on the USB serial connection between the node and the PC, arriving frames can cue up in the sensor node, hence the timestamp given by the PC is not accurate. In the future we want to let the sensor node itself give the timestamp.

Figure 4 represents a screenshot of wireshark, showing the decoded field of the MAC header. In this case no IP packet was transported in the frame.

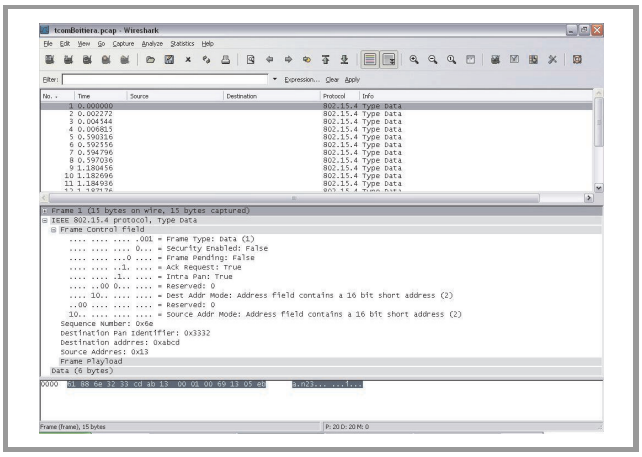

Fig. 4. Screenshot of wireshark, showing the MAC header.

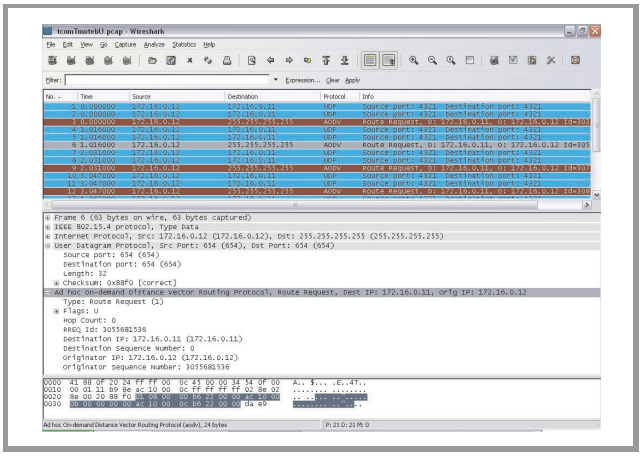

Fig. 5. Screenshot of wireshark, showing an AODV route request message.

Figure 5 shows an AODV route request message, encapsulated in an UDP/IP packet, transported by an IEEE 802.15.4 frame. All details of the captured frames, on any layer, can be decoded and analysed, which makes this tool very interesting for debugging protocols or applications running on the wireless nodes.

4. Effective data capacities

Due to the MAC protocol (unslotted CSMA-CA) and the possible multiple hops between source and sink, the effective data capacity will always be smaller than the data rate at the physical layer. In view of developing applications on a MANET or WSN based on IEEE 802.15.4, it is interesting to have an idea what the maximum data throughput could be, using this given protocol. In this section, we calculate the theoretical effective data capacity for a single- and multi-hop scenario and compare this with measurements on a real network. A similar study was conducted in [4], although not under the same conditions and using the same tools.

In the following, the effective data capacity is defined as the maximum achievable data rate for a user application, in the absence of any cross traffic. All calculations and experiments are performed under the following conditions: the nodes are configured in the IEEE 802.15.4 compliant beaconless mode, supporting an over the air data rate of 250 kbit/s at the physical layer (C_{PHY}), short addresses are used, the optional acknowledge frames are enabled and the backoff exponent *BE* is initiated to 0. Further, the nodes will be put in an ideal multi-hop forwarding chain, as represented on Fig. 8. This means that all nodes have the same maximum transmission range $R_{\max}$ and the fourth node in the chain, i.e., node *D*, will not sense an ongoing communication between node *A* and *B*.

Note that in the standard [1] durations are often expressed in number of symbols and not in seconds. In the 2.4 GHz PHY layer duration of 1 byte $= 2$ symbols $= 32\,\mu$s.

4.1. Theoretical approach

In a first step we will calculate the effective data capacity for a single-hop connection between 2 neighbours. To allow the MAC layer to process the data received by the PHY, each data frame is followed by an interframe spacing (IFS). If the length of the MAC protocol data unit (MPDU) is larger than 20 bytes, a long IFS (LIFS) of 640 μs will be used as shown in Fig. 6. The spacing T_{ack} between a data frame and the acknowledgement (ACK) frame equal the TX-to-Rx maximum turnaround time ($= 192\,\mu$s). Both LIFS and T_{ack} have been measured by a communication analyzer and the values given by the standard are respected by the CC2420 radios on the Tmote Sky. To calculate the upper bound of the single-hop effective data capacity *C*, the length of the MPDU is set to its maximum, i.e., 127 bytes. The size of the ACK frame is always 11 bytes. As *BE* is initialized to 0 and there is no cross traffic, there will be no backoff delay in this scenario.

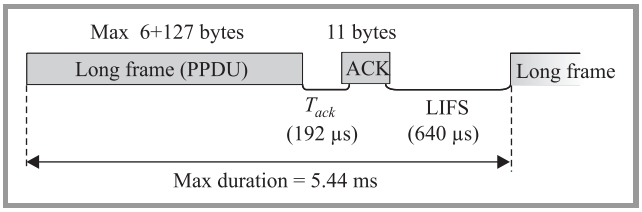

Fig. 6. Long inter frame spacing.

Note also that all other delays like CCA time and turnaround time are included in T_{ack} and LIFS. Hence the total time between 2 long data frames T_{tot} is given by

$$T_{tot} = T_{long\ frame} + T_{ack} + T_{ack\ frame} + LIFS = 5.44\,\text{ms} \quad (1)$$

with $T_{long\ frame} = 133 \cdot 32\,\mu$s, the time it takes to send out a long frame of 133 byte, and $T_{ack\ frame} = 11 \cdot 32\,\mu$s.

Figure 7 shows the details of a data frame of maximum size. The frame consists of 5 bytes synchronization header (SHR) and 1 byte physical header (PHR). On the MAC layer there are, using short addresses, 9 bytes of MAC header (MHR) and 2 bytes of frame check sequence (FCS) (CRC16). On the network layer, there is a 20 byte IP header and an 8 byte user data protocol (UDP) header. This leads to a total overhead of 45 bytes, meaning there are only 88 bytes left for user data.

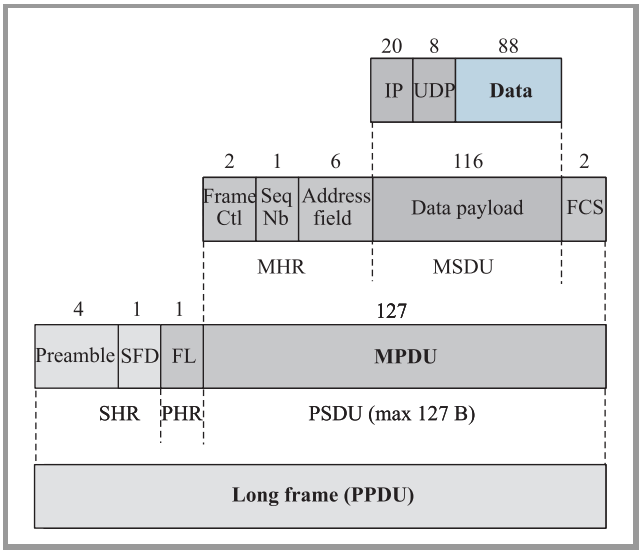

Fig. 7. Structure of an IEEE 802.15.4 data frame. Explanations: MSDU – MAC service data unit, PSDU – PHY service data unit, PPDU – PHY protocol data unit.

Taking into account the MAC layer and the protocol overheads, the theoretical maximum throughput that a single-hop transmission can achieve is given by

$$C = \frac{T_{user\ data}}{T_{Tot}} C_{PHY} = 129.41\ \text{kbit/s} \quad (2)$$

with $T_{user\ data} = 88 \cdot 32\,\mu$s, the time it takes to send the user data over the PHY interface and $C_{PHY} = 250$ kbit/s. Hence,

the theoretical upper bound of the effective data capacity available for the user is only 52% of the PHY data rate.
In a multi-hop scenario with N nodes ($N \leq 4$) and in the absence of the backoff mechanism, the upper bound of the effective data capacity is given by

$$C/(N-1)\,, \tag{3}$$

since only one of the N nodes can transmit at any time. In case of an ideal forwarding chain for $N > 4$ (Fig. 8), the 4th node can transmit in parallel with the first, without interference, leading to an effective data capacity of $C/3$ for any $N > 4$.

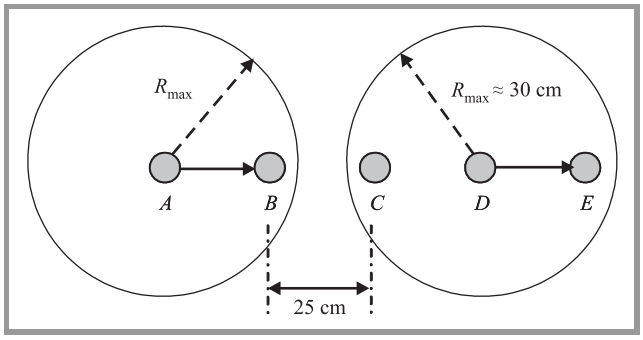

Fig. 8. The ideal forwarding chain.

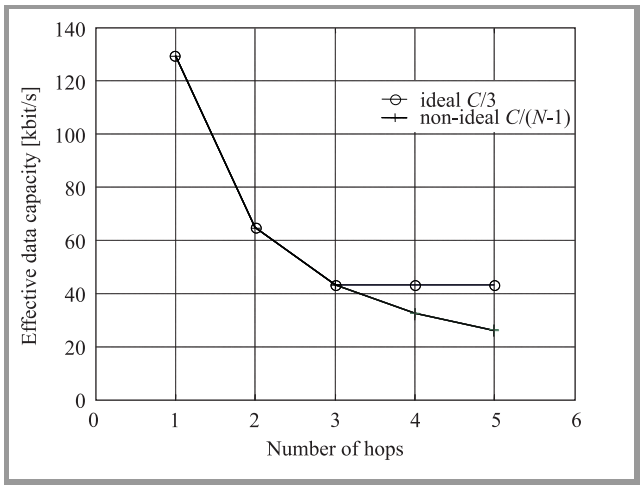

Fig. 9. Upper bound of the theoretical effective data capacity in an ideal and non-ideal forwarding chain.

In a non-ideal multi-hop scenario, with the N nodes in each other's interfering zone, the effective data capacity will still be governed by Eq. (3). Figure 9 presents the upper bound of the theoretical capacity in an ideal and non-ideal ad hoc multi-hop forwarding chain.

4.2. Experiments

Experiments are performed with the Tmote Sky modules under the same conditions as the theoretical calculations. The transmission power of the nodes is set to the minimum, resulting in a transmission range of about 30 cm. The nodes were placed on a straight line at intervals of 25 cm.
The application software running on the nodes is very simple. For the single-hop scenario, node B sends an UDP packet with 88 bytes of data, waits for a given time T_{wait}, sends the next packet and so on. Node A resets a timer, waits for 1000 received packets, gives a timestamp and reports to a PC. By fine tuning T_{wait}, a maximum is achieved. For a 2-hop scenario, node C is the one sending the UDP packets, and node B just relays the packets to the destination node A, etc.
Figure 10 shows the results of the measurements for a single- and a multi-hop scenario up to 4 hops. In all cases the measured data capacity is less than the expected data capacity, e.g., for the single-hop scenario 101 kbit/s is measured instead of the expected 129.41 kbit/s (Eq. (2)). The main reason for the discrepancy is due to Contiki and how it is implemented on the Tmote Sky module. The CC2420 radio module of the source node, node B in the single-hop case, will empty its transmission buffer after reception of the ACK frame. From that moment, the MSP430 microcontroller can transfer the next MAC frame to the radio module. This is done via an SPI interface, connecting the microcontroller to the CC2420 radio. Unfortunately in the OS Contiki, the baud rate of this SPI is set too low, and the transfer of the 127 bytes over the SPI takes more than the minimum time LIFS between 2 frames. As a consequence, the total time between 2 frames is more than the predicted 5.44 ms (see Fig. 6). In a multi-hop scenario the situation is even worse. First of all there will be collisions on the air interface, hence the backoff mechanism will be activated. Further, in a relaying node, the MAC frames have to travel twice over the slow SPI interface and the IP packets have to be processed by the microcontroller.

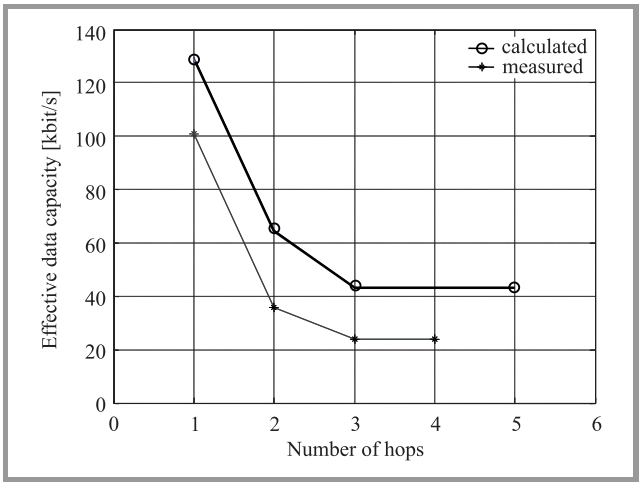

Fig. 10. Theoretical and measured effective data capacity in case of an ideal forwarding chain.

The measured effective data capacity of a 3-hop chain and a 4-hop chain are the same. This validates the assumption of a $C/3$ data capacity for an ideal chain in case of $N > 4$.

5. Positioning based on RSSI

To exploit the data coming from the sensors, it is often inevitable to have an idea of the (relative) position of the sensor nodes in the network. Equipping the nodes with a GPS module could be a solution, although this implicates an extra antenna on the node and a clear view of the sky, which is not always feasible. Furthermore, a GPS module will increase the price of a node and will compromise the battery lifetime.

Some other well documented techniques for retrieving the position of the nodes in a wireless network are based on radio hop count, RSSI, time difference of arrival or angle of arrival. A good overview presenting the most important localization techniques can be found in [5]. A relative simple technique is the one based on the RSSI, also called radio positioning. In this technique the nodes look at the power of the received signal from their neighbours and try to estimate the distances to their neighbours for localization. In the IEEE 802.15.4 standard, the radio receivers are bound to measure the received signal strength of arriving frames, hence the choice for using this technique.

The technique of radio localization is well described in literature and practical evaluations of the method have been presented. Mostly the method is found inaccurate, only in open outdoor environments reasonable results can be obtained [6]. To gain some practical experience on the accuracy of the method, we decided to implement the positioning based on RSSI on our WSN and to do some basic field tests.

5.1. Propagation model

A necessary condition in this technique is to use a good propagation model. For this experiment, the transmission channel was intentionally kept very simple, with only a ground reflection and no other obstacles or fading sources present. In a wireless environment, the received signal strength may be expressed as

$$P_{Rx} = P_{Tx} + G_{Tx} + G_{Rx} + L, \tag{4}$$

where P_{Tx} is the transmitted power, G_{Tx} and G_{Rx} are the transmit and receive antenna gains and L is the path loss in dB. In free space, the path loss of the transmission channel is governed by a $1/r^2$ power-law. The presence of the ground between the antennas however, allows a second ray to reach the receiving antenna. As the receiving antenna moves away from the emitting antenna, the two rays add successively constructively and destructively, giving rise to oscillations around the $1/r^2$ power-law. At a distance

$$d >> \frac{4\pi h_{Tx} h_{Rx}}{\lambda} \tag{5}$$

from the emitting antenna the oscillations around a $1/r^2$ power-law disappear and are replaced by a $1/r^4$ power-law [7], as shown in Fig. 11.

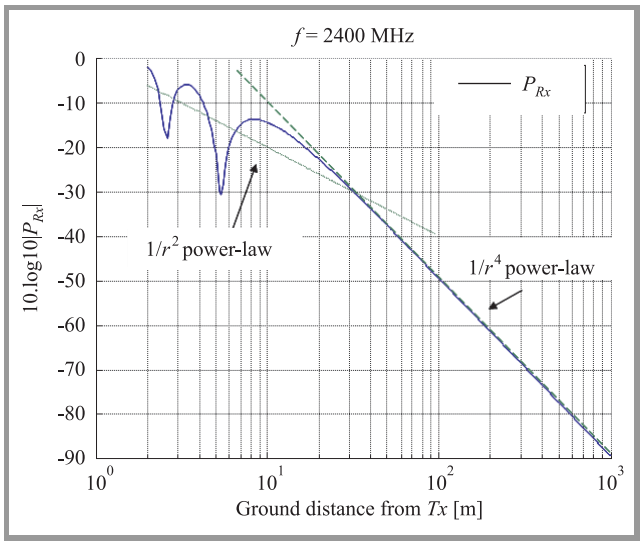

Fig. 11. Simulation of the received power for the 2-ray model.

5.2. Experiments

To avoid fading as shown on Fig. 11, we decided to limit the height of the antennas to 25 cm above the ground, which seems to be a realistic height for a real implementation. In this case, the oscillations due to multi-path fading will disappear for $d > 6$ m, leading to a smooth $1/r^4$ power-law for the path loss. In a first experiment a calibration was performed. This calibration also allows to verify the $1/r^4$ power-law and gives an idea of the ranging capability of the method.

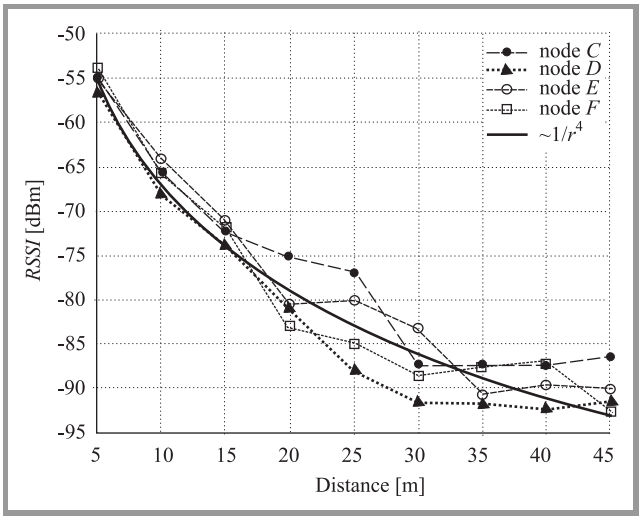

Fig. 12. Calibration measurements for 4 different nodes, confirming the propagation model.

Figure 12 shows the result of the calibration for 4 different nodes (nodes *C-D-E* and *F*). The receiving node was displaced from 5 to 45 m in steps of 5 m. The bold solid line represents a $1/r^4$-curve fitted over the measured data, serving as a reference. A first conclusion can be drawn here. The $1/r^4$ propagation model is confirmed, but

the ranging error increases over distance. This increasing error depends on both noise and attenuation rate [6]. The $1/r^4$-curve flattens out, meaning that a slight error in the measurement of the *RSSI* will lead to a large ranging error, in some cases up to 30% of the actual range. Note also that the accuracy of the *RSSI* measurement by the CC2420 is only ± 6 dB [3].

In a second experiment, 4 anchor nodes (nodes *C-D-E-F* of the previous experiment) were placed in the 4 corners of a half-football field. A fifth node was displaced at 20 different locations in the field logging the *RSSI*-values of the anchor nodes. For each position and anchor node at least 10 values are measured for averaging. Off-line, the distance to each anchor node was retrieved and the position was calculated using a range-based least-squares multilateration method.

Figure 13 shows the result for the 4 corners of the penalty area. The retrieved positions are indicated by the arrows. The median localization error in this experiment was 17 m and a 90th percentile of 26 m.

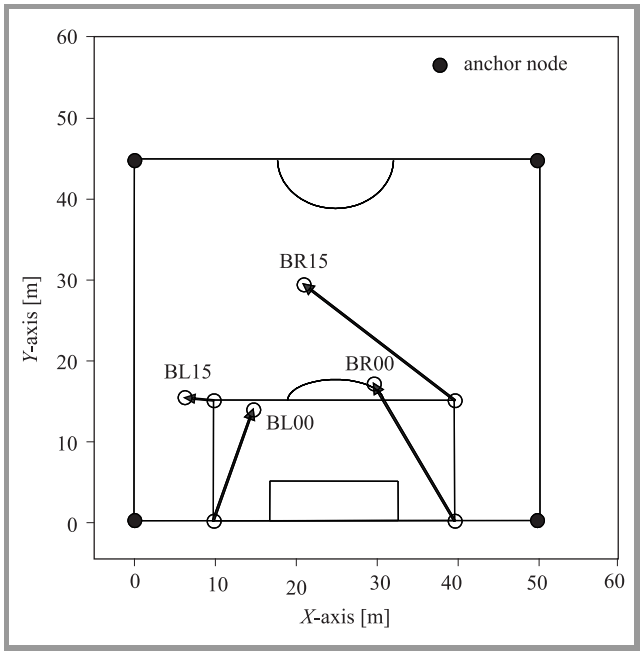

Fig. 13. Experimental results of RSSI-based positioning on a half-football field using 4 anchor nodes.

Although the results seem inaccurate the method was found out to have some potential and improvements to enhance the accuracy can still be introduced. A first possible improvement could be the use of external omni-directional antennas instead of the internal antennas. A second improvement could be the reduction of the test area, so that the distances between the nodes and the anchor nodes will decrease, leading to better ranging performance. For the moment only the *RSSI*-values to the anchor nodes are used to calculate the position. Using also the *RSSI*-values to other nodes and a network compensation based position computation method, will further enhance the accuracy. In the future more experiments will be conducted implementing these enhancements and evaluating also the radio localization in less optimal outdoor conditions like environments with vegetation and trees.

6. Conclusions

The research on ad hoc networks and WSN recently started at the Belgian Royal Military Academy. In this paper a summary was given of some the first ongoing activities in this domain. The work is not only focussing on a theory and simulations, but also practical implementations are considered. To do so, an IEEE 802.15.4-based WSN is used. The RTOS running on the nodes is Contiki, the network layer is IP-based and AODV is used as ad hoc routing protocol. To be able to debug applications on the IEEE 802.15.4-based wireless network, we developed a packet sniffer which can be integrated in wireshark. A plug-in was written for wireshark, as the IEEE 802.15.4 standard was not yet supported.

In view of future applications on the wireless network, a theoretical study of the effective data capacity was made and compared with measurements performed on the sensor network. For a single-hop scenario, the theoretical upper bound of the effective data capacity available for the user is only 129.41 kbit/s or 52% of the PHY data rate. In practice, due to the OS Contiki and how it is implemented on the wireless sensor nodes, the available effective data capacity is even less.

To exploit geographically meaningful sensor data, it is inevitable to know the (relative) position of the sensor nodes in the network. A simple technique is the one based on the RSSI. Mostly this method is found inaccurate, and only in open outdoor environments reasonable results can be obtained. We performed some experiments of positioning based on RSSI on a half-football field. The median localization error was 17 m. The method has some potential in outdoor environments and further improvements to achieve better accuracy will be introduced.

References

[1] "IEEE Std 802.15.4-2003", IEEE 802.15 WPAN™ Task Group 4, 2003, http://www.ieee802.org/15/pub/TG4.html

[2] "Tmote Sky datasheet", Moteiv corporation, 2006, http://www.moteiv.com/products/

[3] "CC2420 datasheet, Chipcon products from Texas Instruments", Texas Instruments, 2004, http://www.chipcon.com/

[4] T. Sun, L.-J. Chen, C.-C. Han, G. Yang, and M. Gerla, "Measuring effective capacity of IEEE 802.15.4 beaconless mode", in *Wirel. Commun. Netw. Conf. WCNC 2006*, Las Vegas, USA, 2006, pp. 492–498.

[5] I. Stojmenović, *Handbook of Sensor Networks, Algorithms and Architectures*. Hoboken: Wiley, 2005, Chapter 9.

[6] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization", *ACM Mob. Comp. Commun. Rev. (MC2R)*, Special Issue on Localization Technologies and Algorithms, 2007.

[7] T. S. Rappaport, *Wireless Communications, Principles and Practice*, 2nd ed. Upper Saddle River: Prentice Hall, 2002.

Bart Scheers was born in Rumst, Belgium, in November 1966. He obtained his degree of engineer, with a specialization in telecommunications, at the Royal Military Academy in 1991. After his studies he served as an officer in a territorial signal unit of the Belgian Army. In 1994 he became an Assistant at the Royal Military Academy in the field of signal processing. In 2001 he presented his Ph.D. thesis on the use of ground penetrating radars in the field of humanitarian demining. From 2000 he works as an Associate Professor in the Telecommunication Department. His current domains of interest are ad hoc networks, wireless sensor networks and software defined radio.

e-mail: bart.scheers@rma.ac.be
Royal Military Academy
Department CISS
Renaissancelaan 30
B1000 Brussels, Belgium

Wim Mees was born in Sint-Truiden, Belgium, in November 1967. He obtained his engineering degree with a specialization in telecommunications at the Royal Military Academy in 1990. After his studies he served as an officer in a signal unit of the Belgian Army working in support a NATO HQ. In 1992 he returned to the Royal Military Academy as a lecturer and he currently works as an Associate Professor in the Communication, Information System and Sensors Department. In 2000 he presented his Ph.D. thesis on the use of artificial intelligence for the semi-automatic interpretation of high-resolution satellite images. His current domains of interest are command and control systems and information security.

e-mail: wim.mees@rma.ac.be
Royal Military Academy
Department CISS
Renaissancelaan 30
B1000 Brussels, Belgium

Ben Lauwens was born in Leuven, Belgium, in January 1977. He got his degree of engineer, with a specialization in telecommunications, from the Royal Military Academy in 2000. After his studies he served as an officer in a territorial signal unit of the Belgian Army. In 2005 he obtained a Master in engineering at the Katholieke Universiteit Leuven. From 2005 he works as an Assistant in the Telecommunication Department of the Royal Military Academy and is working on a Ph.D. thesis entitled “Hybrid packet-event/fluid-flow network simulation with applications to communication and wireless sensor networks”.

e-mail: ben.lauwens@rma.ac.be
Royal Military Academy
Department CISS
Renaissancelaan 30
B1000 Brussels, Belgium