

AEWSN: ACADEMIC EDUCATIONAL WIRELESS SENSOR NETWORK

Karl Benkič, Marko Malajner, Žarko Čučej

Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Maribor, Slovenia

Key words: Wireless sensor networks, AeWSN, modeling, simulating, evaluating

Abstract: Wireless sensor network (WSN) is a collection of wireless sensor nodes with wireless communication capability which is often used for military application, real-time surveillance, and similar time-critical applications. This article offers an overview of a wireless sensor network system called AeWSN. AeWSN is a custom designed WSN with software and hardware support. The proposed system was developed as a byproduct of PhD studies and is intended for use as an academic wireless sensor research platform, offering some sort of a framework for higher (level 3 or higher) ISO/OSI layer development in WSN. It offers network and protocol modeling, developing, simulating and experimental testing as well as evaluating it. Finally we demonstrate AeWSN on a simple application.

AeWSN: akademsko izobraževalno brezžično senzorsko omrežje

Ključne besede: Brezžična senzorska omrežja, AeWSN, modeliranje, simuliranje, ocenjevanje

Izvleček: Brezžična senzorska omrežja (BSO) so postala zanimivo raziskovalno področje v preteklih nekaj letih. Zaradi določenih razlik s podobnimi tipi omrežij (Ad-hoc, MANET) raziskave potekajo dejansko na vseh plasteh referenčnega ISO/OSI modela.

Za testiranje ali razvoj novih algoritmov za uporabo v brezžičnih senzorskih omrežjih je ključnega pomena, da le te najprej simuliramo v simulacijskem okolju, kasneje pa še ovrednotimo in testiramo na realnem brezžičnem senzorskem omrežju. V ta namen je potrebno najprej izbrati dobro simulacijsko okolje in znotraj le tega razviti ustrezen model brezžičnega vozlišča. Obstaja več simulacijskih orodij za simuliranje brezžičnih senzorskih omrežij (OMNET++, OPNET, GloMoSim,...), vsako od njih pa ponuja določene pozitivne in negativne lastnosti. Kljub razširjenosti simulacijskih orodij v bistvu ne obstaja nobena širša študija, ki bi opredelila uporabnost ali prevlado enega od teh. Med izbiri simulatorja za BSO smo najprej izbrali dva potencialno nam najbolj priročna orodja: OPNET in OMNET++. Izberi OPNET-a je so botrovale izkušnje, ki jih z orodjem imamo in dejstvo, da OPNET ponuja (sicer zaprti) model senzorskega vozlišča ZigBee tehnologije. Za razvoj svoje lastne platforme smo se odločili predvsem zaradi preprostega spreminjanja strojne opreme po potrebi in zaradi pridobivanja znanja in izkušenj, ki ga to delo prinaša. Razvito senzorsko vozlišče SPaRCMosquito med drugim omogoča tudi zamenjavo strojne programske opreme na daljavo, kar velja za eno najbolj priročnih lastnosti vozlišča. Kljub veliki ponudbi programske opreme pa smo zasledili pomanjkanje take, ki omogoča modeliranje BSO in povezovanje sistemov v celoto. Tako je nastala AeWSN (Academic education wireless sensor network) platforma, ki združuje programsko in strojno opremo ter model simulatorja.

AeWSN je sestavljen iz treh glavnih sklopov: Strojne opreme (SPaRCMosquito), programske opreme (SPaRCSoft) in simulacijskega modela (SPaRCSimMod). SPaRCMosquito senzorski modul temelji na Cortex 1768 mikroprocesorju in MRF24J40 radijski enoti. Senzorski modul omogoča zamenjavo strojne programske opreme na daljavo ter uporabo različnih senzorjev združljivih z AD, SPI, CAN in I²C vodilom.

SPaRCSoft je skupek programske opreme: strojne programske opreme, ki teče na SPaRCMosquito modulu in SPaRCSoft programa (Fig. 4), na katerem je programska oprema, ki kontrolira SPaRCMosquito bazno postajo (BS). SPaRCSoft omogoča, da se SPaRCMosquito obnaša kot bazna postaja, navadna senzorska enota ali vohiljač. SPaRCSoft omogoča modeliranje in realno časovni prikaz stanja omrežja (Fig. 5 in Fig. 6), kjer s pomočjo lokalizacije izrisujemo senzorska vozlišča in v določenem razmerju glede na realno omrežje. Delovanje AeWSN sistema smo preizkusili v par različnih aplikacijah, med drugim tudi na problemu določanja lokacije posameznih senzorskih vozlišč. .

1. Introduction

Wireless sensor networks has become intensive research field recent years. Due to their difference from conventional Ad-Hoc wireless networks, researches have covered all the layers of an ISO/OSI model. Originally, IEEE as the Working Group for Wireless Personal Area Networks (WPANs) defined the 802.15 standard, and soon the 802.15.4 standard known as low-rate wireless personal area networks was defined. ZigBee alliance /1/ built their wireless sensor network solution called ZigBee on top of the 802.15.4. standard, as used for physical and medium access layer.

Primarily the need for wireless sensor networks arose in army applications. Remote battlefield and Base-Camp surveillance became simpler and possible using wireless technology. In the future, military applications and decisions

will be based more and more on data gathering, including data from the battlefields. WSN is one of the most promising intelligence data gathering systems /2/.

A typical WSN consists of a large number, even a few thousand, nodes equipped with different types of sensors, microprocessors, radio transceivers, and usually an autonomous power supply, deployed randomly within the area of interest with the goal of collecting sensors measured data and transferring it through one or more data sinks called User Access Point (UAP) or a Base Station (BS) to the end users of WSN services. The random deployment of sensors, for example in battlefields or in plantations, usually means that the positions of the nodes are neither known exactly nor accessible for maintenance and local administration. Consequently, they should have the capabilities for self-organization, within an ad-hoc multihop

network fashion. Under normal working circumstances the majority of traffic flows towards BS /3/, since the BS is a data collector.

1.1. Problem formulation

When developing new algorithms, it is crucial to test and evaluate them. Testing and evaluating is possible through two phases: simulating and evaluating the algorithms in the simulators and then testing, and evaluating algorithms on a real WSN. When simulating the WSN, it is crucial to have a good wireless sensor network and node model and, of course a suitable simulation tool.

There are several network simulators (NS2, OMNET++, OPNET) to choose from but mainly none of them supports WSN natively. OPNET /4/ supports practically all physical radios with many types of modulations and, therefore, it is quite easy to implement a custom PHY layer for wireless communications. Among other models OPNET supports ZigBee and all three types of devices (RFD, FFD and Co-ordinator). For PHY and MAC layers there is source code available, but for the network and application layers there is only an object code. OMNET++ on the other hand, offers a good discrete simulator and 3 party models for the WSN simulations /5/ with or without the limited support, what is expected, because the majority of simulation models are open source projects, done by non-profit organizations (faculties) or individuals. The authors in /6/ offered answers which simulator is more appropriate for certain types of simulation (for example: they recommend the NS-2 simulator for high-precision PHY layers simulation) and how a simulator differs in the same setup and scenarios. The same comparison was done in /7/ where the authors found great differences in MANET simulations executed in the OPNET Modeler /4/, NS-2 and GloMoSim. For example: OPNET simulates a very high time delay in MANET compared to NS-2 and GloMoSim /7/. After evaluation the OPNET simulation tool was selected for simulating WSN.

When selecting the hardware evaluation platform a decision had to be faced: to use one of the existing platforms for wireless sensor networks or develop a new one, despite a few solutions available on the market. AeWSN is developed for education and research work, which at some point include working with images, so the WSN node should use a CPU which is powerful enough. We didn't want to depend on

only one CPU, sensor(s) and radio modules; so, the node has to have modular design. Different setup properties and experiments led us to believe that custom hardware would be the smart choice for us. Hardware choice was based on our previous knowledge, and support from different vendors and companies.

The third part of the problem was the appropriate support software selection. Because there is practically no similar software for developing protocols and evaluating WSN parameters it was decided to create one.

1.2. Related work

There are a lot of manufactures who provide the market with solutions for wireless sensor networks. Some manufactures created a consortium called ZigBee Alliance /1/. Zigbee alliance proposed high layer communication protocol based on IEEE 802.15.4 standard for wireless personal networks (WPAN). The main advantage of a ZigBee is its simpler and cheaper protocol in comparison with Bluetooth, which makes ZigBee very appropriate for wireless sensor networks. Two alliance members, Microchip and Texas Instruments are the world's best-known providers for ZigBee protocols and equipment. As a member of the ZigBee Alliance, Microchip offers its customers a certified ZigBee Compliant Platform (ZCP) for the ZigBee® PRO, ZigBee RF4CE, and ZigBee 2006 protocol stacks. The ZCPs ensure interoperability with the ZigBee industry standard. Microchip's ZigBee Compliant Platforms consist of the 2.4 GHz IEEE 802.15.4 compliant MRF24J40 transceiver products, the PIC18, PIC24, dsPIC, and PIC32 families of microcontrollers, and the certified firmware protocol stacks. TI offers cost effective low power RF solutions for the ISM band, ZigBee hardware and software solutions. TI's product offers include RF transceivers, RF transmitters, RF receivers, System-on-Chips, Front Ends and ZigBee Processors.

Some universities and companies use basic hardware to produce their own wireless sensor nodes but, in general, almost all of them use IEEE 802.15.4 compliant hardware.

Table 1. shows the that market offers WSN nodes ranging from simple low power CPUs to high performance ARM processors (PXA271). A conclusion can be made that the selection of a WSN node is typically depended on application.

Table 1: The comparison of some WSN nodes available on the market.

	WeC Mote	Mica2 Mote	MiCAz	Medusa MK2	μAMPS	Imote2.NET
Radio	TR1000	CC1000	CC2420	TR1000	LXM3162	CC2420
MCU	AT90LS8535	ATMEGA128L	ATMEGA128L	ATMEGA 128L AT91FR4081 ARM	StrongARM SA-1100	Marvell PXA271
Operating system		Tiny OS	TinyOS	uCOS-II	eCos	Win CE, .NET micro framework
Memory	23KB EEPROM	128 + 512K FLASH	128 + 512K FLASH	1MB FLASH		32MB FLASH 32MB SDRAM

2. AeWSN.NET system

2.1. Hardware – SPaRCMosquito

The Laboratory for Signal Processing and Remote Control (SPaRC) has developed its own wireless sensor node platform called SPaRCMosquito. One of the reasons for developing its own platform was the absence of an ARM microcontroller based sensor node platform on the market. Microcontrollers based on ARM architecture are very popular these days for mobile devices due to their small power consumption, efficiency, reasonable prices, etc. Two sensor node platforms have been developed. One for connecting to a PC, used for data gathering (base station), which is named the SPaRCMosquito Base Station. The independent sensor node, deployed for the field (sensor node), is called SPaRCMosquito. Microchip's MRF24J40 radio module /8/ based on 802.15.4 standard and TI's CC2500 radio module /9/ were chosen for the wireless part.

2.1.1. SPaRCMosquito Base Station

The task of a base station is the gathering of data from an entire wireless sensor network, so, the connection to the higher entity is needed. The SPaRCMosquito Base Station can communicate with a PC, laptop, etc. via USB link. The heart of base station is a low power LPC 1768 Cortex, 32-bit microcontroller. In addition to the microcontroller, the platform consist of USB and LED drivers, JTAG connector, general-purpose connector (includes I²C, SPI, ADC and GPIO) and other. The power supply is assured over the USB connection. Fig. 1 presents the SPaRCMosquito Base Station.

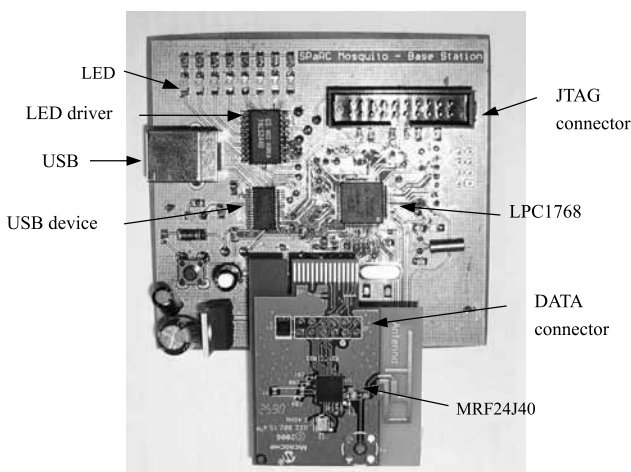


Fig. 1: SPaRCMosquito base station

2.1.2. SPaRCMosquito Sensor Node

The SPaRCMosquito Sensor Node is based on the same platform as the base station with some modifications. There is no longer USB driver and some LED indicators. Instead a connector for the sensor (for measuring physical phenomena), a DC/DC converter and flash memory have been added (Fig. 2).

MRF24J40 Daughter board

The MRF24J40 Daughter board from Microchip is a wireless radio originally designed for the PICDEM Z evaluation platform. This module was used for the SPaRCMosquito platforms. MRF24J40 is fully compatible with IEEE802.15.4 (ZigBee and MiWi). Communication between radio module and microcontroller is established over a 4-wire SPI protocol. CSMA-CA, automatic acknowledge and retransmit are implemented physically on a chip. The data rate is up to 250 kb/s with QPSK modulation and center frequency 2,4 GHz.

CC2500 Radio Module

The CC2500 is a low-power, low cost radio transmitter from Texas Instruments. CC2500 is very appropriate for wireless device like PC mouse, etc. Its embedded modem is designed for a few different modulations: MSK, 2-FSK, GFSK, OOK) and programmable data rates up to 500 kb/s. Power consumption in the receiving mode is 13,3 mA. The surrounding temperature can be measured using a build-in analog temperature sensor. In order to implement the CC2500 radio module only a chip was purchased, because only a chip, few extern passive elements and folded copper dipole antenna on substrate is needed for the assembly /10/.

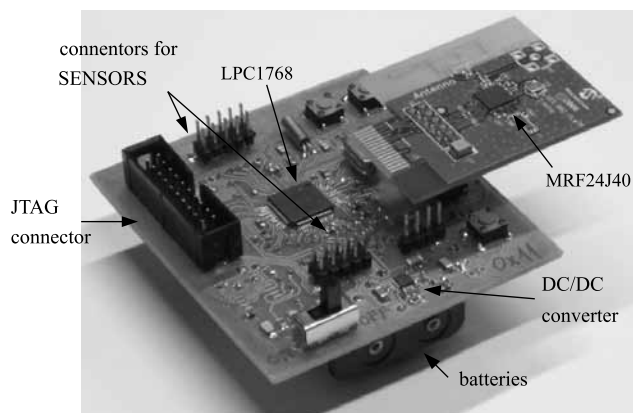


Fig. 2: SPaRCMosquito sensor node

2.2. Software - SPaRCSoft

As mentioned in section 1, wireless sensor networks use at least one base station. A base station is computationally more powerful, with more available resources than standard sensor nodes. In the case of using centralized algorithms (e.g. topology, routing, addressing), the base station normally calculates them. The base station is not normally battery powered, so the energy consumption limitations for the BS do not stand.

In a typical sensor network application, the sensor nodes deployed in the field have low CPU power, limited resources and most importantly: limited power supply. The majority of algorithms (MAC, topology, routing) for WSN are designed for preserving battery power.

Frame	Time(us)	Len	MAC Frame	Seq	Dest	Dest	Source	Invalid Data	FCS
	+1816512		Control	Num	PAN	Addr	Addr	0x25 0x25	RSSI Corr CRC
000001	=1816512	15	0x8841	0x05	0x1234	0x0005	0x1111	0x25 0x25	0xEB 0x69 1

Fig. 3: Data frames used in AeWSN.NET

Software covering the AeWSN.NET can be divided into three main parts:

- Software for the SPaRCMosquito (also referred as node firmware).
- Base station software, including BS firmware and communication protocol to the PC.
- SPaRCSoft, base station and utility software.

SPaRCSoft and all other program components are still the subject of on-going research. Because of this, new modules and algorithms are added weekly.

Wireless communication in AeWSN.NET is based on IEEE 802.15.4 standard /11/. All sensor nodes (including base stations) are using the MRF24J40 module for communication. MRF24J40 is a ZigBee compliant transceiver but, in our case, only data frames with short addressing fields are used (Fig. 3) /11/.

For developing localization, topology, and routing algorithms, the ZigBee or the MiWi stack is too large and complex; therefore we implemented our own, simple algorithms.

2.2.1. SPaRCMosquito firmware

The SPaRCMosquito is, as mentioned in section 2.1.2, based on Cortex 1768 CPU (ARM7 architecture). The programming of a processor is done in ANSI C language with the use of KEIL development tools. The main parts of the software are drivers: for communication with the radio MRF24J40, sensor part drivers, boot loader (a custom boot loader is used for flashing purposes as explained below), and the main application, including topology controls, and routing.

When deployed, the sensor node becomes part of the wireless network. Testing, routing, and topology algorithms on already-deployed networks can be difficult if the firmware can't be replaced remotely. It is for this reason that why we are developing sensor node(s) which can be reprogrammed online (upgraded with any new version of the firmware). If we want to upload new firmware on the sensor node, we must reprogram the microcontroller's flash memory with new firmware. The Internal LPC's flash memory 512KB with external 1 MB memory flash has been extended for this purpose. We have developed an "extended" bootloader which flashes the ARM from firmware on the external memory flash.

New firmware is transferred from base station to each sensor node wirelessly. Each sensor's node stores new firmware in external flash memory. When done, the first byte flashed on the external flash is marked as 0x01 (new firmware). The program running on the ARM resets the

ARM. When entering our own bootloader, first, the check of the first byte on the external flash is made. If the first byte is set to 0x00 the ARM boots normally, otherwise (if the first byte on the external flash is set to 0x01), code in our bootloader copies firmware from the external flash to ARM internal flash, clears the first byte on the external flash, and moves the program counter to the start of the program just flashed.

2.2.2. SPaRCSoft

SPaRCSoft is the main application in the AeWSN.NET system. WSN modeling, testing, and evaluating is done using this application. It is also a gateway to the WSN base station. SPaRCSoft runs on a PC and can be connected to a WSN base station via USB /12/, /13/. The software is modularly based, and currently consist of eight modules (Fig. 4) working together.

The I/O module is the SPaRCMosquito interface. Currently SPaRCMosquito is connected as a virtual com port (VCP) to the PC. The I/O module frames the data from the application module and sends it towards the base station or receives the data, creates an identical data structure, as seen in Fig. 4 and then forwards it to the application module.

Application module contains the firmware for the base station. Due to firmware, the BS hardware module can act as a base station, regular module, analyzer module (like Microchips Zena), and a sniffer. All higher modules communicate with the application module.

Remote firmware upgrade module reads the binary file using the nodes firmware and distributes it over the network as stated in section 2.2.1.

Web server module is used for accessing data over the internet (e.g. reading temperature or moisture).

WSN properties module uses evaluation metrics for evaluating protocols if necessary.

SPaRCGraph module calculates different topology and routing algorithms and distributes the virtual topology over the network. Data for simulating are available through this module.

Meta data module prepares data and information for simulating in OPNET or for firmware development.

Internal settings for the modules are, if necessary, all read from the XML. Users can change it to XML or in SPaRCSoft GUI, if offered. Some settings (like com port settings) can only be changed only in XML.

The basic structure of the SPaRCSoft is a mathematically defined undirected graph $G = (V, E, W)$ with vertexes (sensor nodes), edges (communication links between them), and weights (function of channel properties, delay and similar). The graph model is used for all the protocol calculations. SPaRCSoft can work online (connected to the real network) or offline (as a modeling and analyzing tool). When working online, sensors can be visualized as a graph on the SPaRCSoft graphic panel. Nodes are randomly distributed and the link shows the nodes neighbors. Of course, the graph can be imported (for example TSPLIB), created by the user or created randomly (working offline). When the basic structure is created (or imported) calculations can be done on it. The majority of the research work is done according to topology and routing protocols, so the SPaRCSoft is prepared for the mentioned topics. From the ISO/OSI perspective the physical and data link layers are implemented on the MRF modem (PHY and MAC). Higher layers (topology, routing) are implemented in the application module.

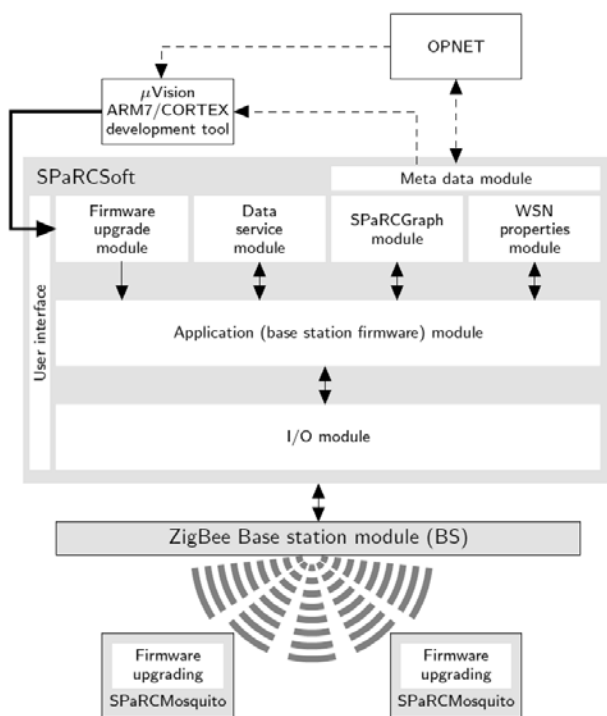


Fig. 4: SPaRCSoft modules

Let's assume that 10 wireless nodes and the base station are in WSN at the moment and the initial phase of creating the network is over. The network topology is shown in Fig. 5. All nodes are transmitting at 100% of their power. The circle around node 10 represents its communication radius without channel properties – the gray color. The communication radius with logarithmic channel model is displayed in the green color. We decided that simple topology routing (minimum spanning tree - MST) would be used. SPaRCSoft firstly calculates MST, after that the software calculates the communication radius reduction for every node that still guarantees the MST topology (Fig. 6). Information about

the topology and routing is flushed over the network and the network can be turned into a running mode. It can be seen that node 2 energy was reduced by 50%, and if the free space propagation model is used /14/ Eq.1, energy saving can be simply calculated.

$$P_r = C_f \frac{P_t}{r^2} \quad (\text{Eq. 1.})$$

Where P_r = received power, C_f = transceiver characteristics, P_t = transmitted power, r = distance.

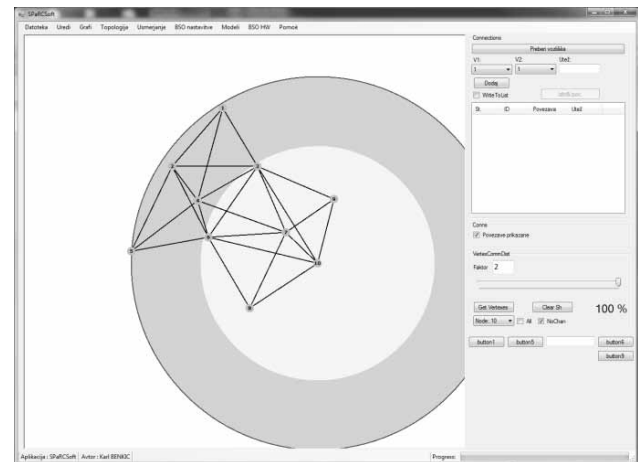


Fig. 5: WSN topology

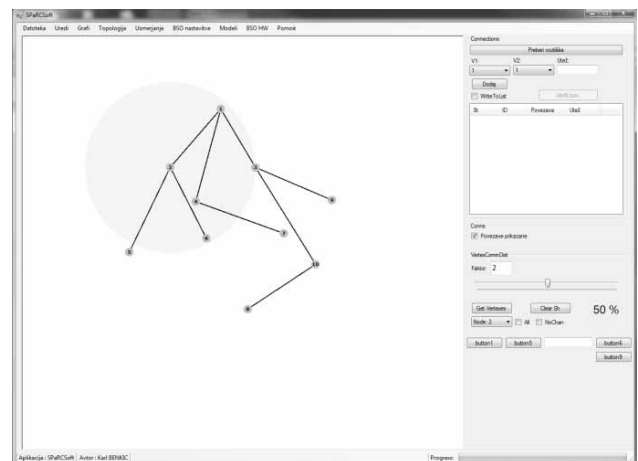


Fig. 6: Calculated "virtual" topology

Topology control is practically done using changing transmission power. If all the MRF module features were be used, the channel number could be used for topology control (in the sense of creating links with different channel numbers). The SPaRCGraph module, in the case of topology control calculates transmitting power of every node that still enables desirable topology (e.g. Connected, MST, RNG, minR) /14/. Energy consumption, from the topology point of view, mainly depends on the transmitting power. Some examples showed us that two hops are less energy-demanding than only one /14/, but achieving a minimum number of hops within a network can sometimes be one of WSN properties goals. This is why we are convinced

that topology control is an important factor in developing WSN routing.

3. Application example - Localization

All data from the sensor's field are usually collected on the base station and this data is usually meaningless without positional information. Therefore, the node's location or position estimation has been a very interesting research topic over the last few years. Position estimation facilitates applications, such as inventory tracking, intruder detection, tracking of fire-fighters or miners, home automation, and patient monitoring. There are many techniques for determining the distances between nodes. Physically, we can measure the time of a signal (acoustic or RF) arrival, the angle of the signal arrival or power of the received signal. Our research focused on measuring the strength of the received signal (RSSI - Received Signal Strength Indicator). The RSSI technique was chosen due to its integration in most 802.11 radio modules. RSSI practically allows us to calculate the received power for each received packet. This parameter can be used together with path-loss and shadowing model for distance estimation.

A centralized algorithm is used for calculating the coordinates of the sensor's nodes. The base station sends RSSI parameters from the entire sensors field (all the nodes) to the PC. Then the PC calculates all the coordinates of the sensors within the network. The packets consist of next data: addresses from two neighboring nodes and RSSI between them. The algorithm calculates the average value of the last 20 measured RSSI and calculates the distances between nodes using one of the appropriate path-loss models (Fig. 7). The first three calculated distances be-

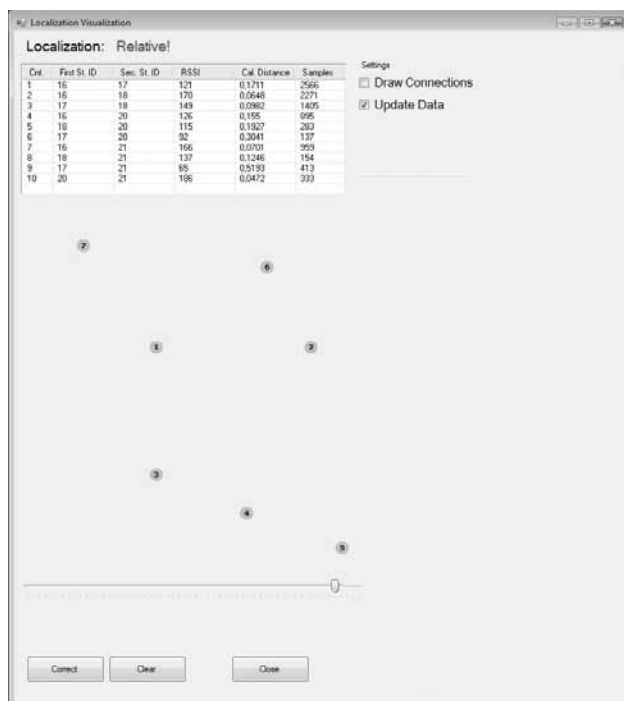


Fig. 7: SPaRCSoft module

tween the nodes creates three reference nodes (anchors or beacons). When we have reference nodes it is easy to calculate other node's coordinates.

3.1. Localization error

During the experiments we discovered that the relative localization error could reach up to 30%. In a typical radio link transmission, waves are reflected and obstructed by all objects illuminated by the transmitter antenna. Calculating the distance within realistic environment is a very complex task. Relative errors occur because of this. Fig. 8 shows relative errors from three different scenarios calculated from ideal models and real measured values / 15/.

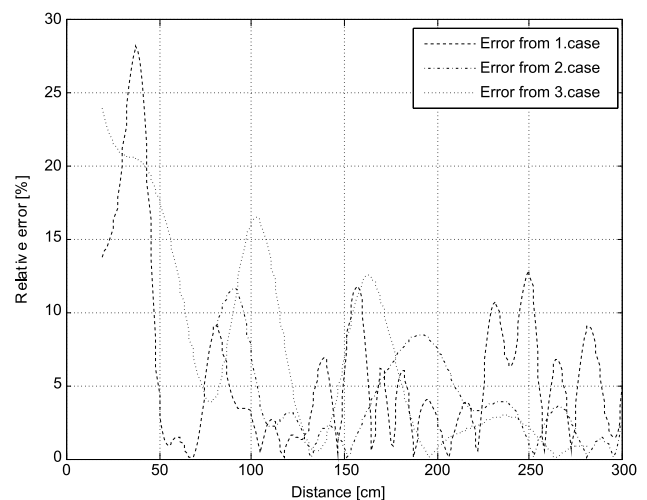


Fig. 8: Localization error

4. Conclusion

In this article, an overview of a wireless sensor network platform called AeWSN is done. Nowadays wireless sensor networking is a very promising, rapidly growing research field. Therefore, many vendors offer wireless transceivers (based on IEEE 802.15.4) or complete wireless sensor nodes (MicaMote, iMOT2...). Despite this, we decide to develop our own wireless sensor node for the AeWSN platform. Developing our own platform gives us the freedom to simply modify the hardware and/or the software to our needs for research or project work, not to mention the experience and knowledge gained during the hardware and software development. AeWSN is still a subject to an ongoing research and development project. Our main goal is to fulfill an AeWNS platform with OPNET simulator node model, which would have the same characteristics as SPaRCMosquito. When the project is stable enough we are planning to launch a web site and making AeWSN a public open-source platform. Finally a simple localization application example is demonstrated as a case study, as to how AeWSN platform works.

Bibliography

- /1/ Z. Alliance. (2010, Mar.) ZigBee Alliance. /Online/. <http://www.zigbee.org/>
- /2/ B. Ames, »Electronics are central to 21st century warfare tactics,« military and Aerospace Electronics, 2004.
- /3/ H. Dai and R. Han, »A node-centric load balancing algorithm for wireless sensor networks,« in Global Telecommunications Conference, 2003.
- /4/ OPNET. (2010, Mar.) OPNET. /Online/. <http://www.opnet.com/>
- /5/ OMNET++. (2010) Simulation Models. /Online/. <http://www.omnetpp.org/models>
- /6/ L. Hogie, P. Bouvry, and F. Guinand, "An Overview of MANETs Simulation," Electronic Notes in Theoretical Computer Science, 2006.
- /7/ D. Cavin, Y. Sasson, and A. Schiper, "On the accuracy of manet simulators," Proceedings of the second ACM international workshop on Principles of mobile computing, pp. 38-43, 2002.
- /8/ Microchip. (2009) MRF24J40. /Online/. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en027752>
- /9/ T. Instruments. (2009) Low Power RF ICs - 2.4GHz - CC2500 - TI.com. /Online/. <http://focus.ti.com/docs/prod/folders/print/cc2500.html>
- /10/ M. Malajner, K. Benkič, and Ž. Čučej, "Implementacija sprejemno/oddajnega modula CC2500 za uporabo v brezžičnih senzor-skih omrežjih," in Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, vol. A, Portorož, 2007, pp. 140-143.
- /11/ D. Gislason, ZigBee Wireless networking. Elsevier, 2008.
- /12/ K. Benkic, M. Malajner, A. Peulic, and Z. Cucej, "Academic education Wireless Sensor Network: AeWSN," in Proceedings ELMAR, Zadar, 2008, pp. 535-538.
- /13/ K. Benkic, M. Malajner, and Z. Cucej, "AeWSN.NET: framework for the wireless sensor networks," in New Master curricula and EU practice : proceedings, Niš, 2008, pp. 91-94.
- /14/ P. Santi, Topology control in Wireless Ad Hoc and Sensor Networks. John Wiley & Sons, Ltd, 2005.
- /15/ M. Malajner, K. Benkič, P. Planinšič, and Ž. Čučej, "The accuracy of propagation models for distance measurement between WSN nodes," in In Proceedings of 16th International Conference on System, Signals & Image Processing, vol. 3, Chalkida, Greece, 2009.

mag. Karl Benkič, mag. Marko Malajner, prof.dr. Žarko Čučej
Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Smetanova 17, 2000 Maribor

Prispelo: 29.03.2010

Sprejeto: 24.06.2011