Energy-Aware Distributed Fence Surveillance for Wireless Sensor Networks

Norman Dziengel, Marco Ziegert, Stephan Adler, Zakaria Kasmi, Stefan Pfeiffer, Jochen Schiller

Department of Mathematics and Computer Science
Freie Universität Berlin, Takustr. 9, 14195 Berlin
{dziengel,ziegert,adler,kasmi,pfeiffer,schiller}@inf.fu-berlin.de

***Abstract*—Fences are used all over the world to protect areas against unauthorized access. While most fences meet these requirements by building a physical and psychological barrier against intruders, this is not sufficient for areas of particular interest like restricted areas of an airport or construction sites with expensive goods. To detect intruders, wireless sensor nodes are integrated into a fence. The requirements for real-world energy-awareness are fulfilled by using active sensors with configurable logic, power saving modes like WOR and power down modes. The sensor nodes are capable of differentiating between several events, with the appliance of a distributed classification algorithm. We present an energy-aware platform for a distributed fence surveillance system in a wireless sensor network. We report on our experience in creating a platform that is concerning energy and performance demands and specifically tailored for the purpose of fence surveillance including fully integrated housing.**

I. Introduction

The financial loss for landlords and owners of construction sites with valuable equipment is assessed as a high economical loss. Therefore a flexible, spontaneously installable, economical and high accurate surveillance system is necessary. A common solution to prevent theft and vandalism on construction sites or to protect valuable goods on airports is a fence. Typically a fence has two disadvantages: Firstly, it reveals the existence of something valuable or interesting and secondly, a fence is simply crossed with some sportsmanship or opened with cheap tools. Current fence surveillance systems still need additional cameras and watchmen to protect borders or entire areas which leads to high costs and complex technical infrastructures [1]. Embedded security systems that are based on distributed event detection with sensor nodes can help to improve and simplify fence installation in the future. Wireless communication brings the advantage of easy installation, but causes problems concerning energy consumption and power supply as well as limited bandwidth. Typical lifetime requirements for WSNs on e.g. construction sites, vary from some months up to several years, depending on the building project. Hence, we need to reduce the amount of energy by applying an intelligent event detection system that is able to perform an in-network evaluation of collected data. Further, an appropriate hardware platform is required that supports numerous energy-saving techniques like Wake-On-Radio (WOR) for the transceiver and suitable power down modes supported by the microcontroller unit (MCU). For an energy-aware surveillance system, all mentioned methods need to be applied to reach the highest lifetime possible. Finally, the whole sensor node needs to be integrated into a fence to provide highest protection to the node itself. To the best of our knowledge, currently no fence surveillance system exists that implements a distributed event detection and supports an energy-aware hardware which can be integrated within the fence.

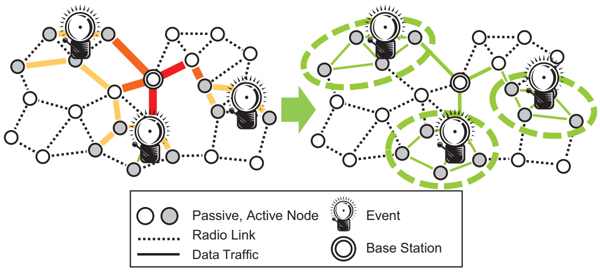

Fig. 1. Left: centralized data evaluation, Right: decentralized data evaluation

We propose a new embedded system with specialized hardware in a full housing that integrates the energy source for a lifespan of more than half a year. The system includes our distributed algorithm for high accuracy event detection, based on distributed versions of classical pattern recognition approaches [2] with an a priori and supervised training. To save energy, we suggest to switch from a centralized to a decentralized data evaluation as depicted in Fig. 1. In the left part of Fig. 1, a traditional Wireless Sensor Network (WSN)[3] uses sensor nodes as data collectors and sends all raw data to the base station (sink). Each node that is used to forward packets towards the sink suffers from the energy costs of additional radio traffic. Nodes that provide the last hop to the sink are frequently used to transmit data to the sink which leads to early failures of these nodes. In the right part of Fig. 1, all calculations are done in the network and only a resulting event may be propagated through the network to the sink. In [4] we presented a feature-based distributed event detection system that assembles and evaluates a combined feature vector within the network to distinguish between multiple events.Based on the event detection, we introduce an energy-aware hard- and software architecture that allows to deploy real-world applications with high life time. This paper contributes to the field of energy-aware distributed fence surveillance for WSNs as follows:

- An energy-aware platform, to cover real-world requirements.
- A System architecture that integrates energy-awareness, event detection and a fully integrated housing.
- Several engineering problems that are often overlooked in testbed deployments are solved.

The remainder of the paper is organized as follows: Section II presents comparable event detection systems. Section III discusses a newly designed sensor node with housing, a multiple base node approach and the distributed event detection system. The energy-awareness achieved by hard- and software adjustments is discussed in Section IV. Experiments and results are covered in Section V. Section VI offers an evaluation of the energy consumption and the communication response time of our event surveillance system. Finally, Section VII concludes the paper.

II. Related Work

Current approaches to integrate event detection in WSNs typically apply a threshold detection, like the fence surveillance system introduced by Kim et al. [1] which does not classify any events. The system is equipped with ground and fence nodes to detect intruders. Combined with a network camera to focus on the intruder, an unmanned ground and air vehicle extend the communication and interaction of the system. The scenario does not cover events that occur in a time-period smaller than 20 seconds, a duty cycle period of 30 seconds is introduced with 10 seconds for sensing and 20 seconds for sleep. With this duty cycle period and the assumption of a maximum available electrical charge of $19\,\mathrm{Ah}$, they calculate a lifespan of approximately 70 days for the ground and the fence nodes.

In [5] Yousefi et al. evaluate an event detection system with one sensor that is attached in the middle of the fence. Yousefi et. al evaluate two different kinds of events (rattling and climbing) that have been exposed to one fence element with a detection rate of 90%. They used a 3D-acceleration sensor for data gathering and event detection. The involved classifier is a Bayesian classifier with an underlying state machine. They extract resonance frequency based features. They are neither introducing a distributed system nor do they disclose the used hardware in detail.

In [6], Genet$^{©}$ offers a fence monitoring system TEDAS^{TM} for military applications. No detailed description of the algorithms are given. All nodes are supplied with a 12 Volt DC wire, and the sink nodes with a 220 Volt AC wire. They make use of a $2.4\,\mathrm{GHz}$ transceiver with 16 Channels, and a 3D accelerometer with a sensitivity of $+/-2\,\mathrm{g}$.

Li et al. [7] research an event detection system for coal mine surveillance to localize collaps events. The Structure-Aware Self-Adaptive principle (SASA) of the 3D environment surveillance is to detect falling nodes or changing positions of sensor nodes by using acceleration data, RSSI evaluations, neighbor loss and some acoustic analysis. They deployed 27 Crossbow Mica2 motes on the ceiling of a coal mine. All nodes have to be set up with an initial known position in the mine. A beacon-based communication is periodically initiated to set up the neighborhood topology of each node. In case of an event, the measured data is mapped with a random hash-function and transformed into a data signature file that is transmitted to the sink. The energy-consumption or life-time is not evaluated.

Our prior work in [8] and [4] which provides a distributed event detection system and reaches a high average detection accuracy of about 87%. Further on, it taught us lessons on how to design a sensor node for the special needs of a fence surveillance system and how integrate it within a fence.

III. AVS-Extrem Event Detection System

The goal of fence monitoring is to distinguish between numerous events like opening the fence, kicking against the fence or climbing over the fence and therefore to prevent the protected area from unauthorized access. In order to monitor these events we place sensor nodes within the fence and additionally on valuable tools like air hammers and vehicles. Each attempted break-in is recognized by several sensors and collaboratively evaluated by the detection system. We need to fulfill the requirements of high computational power, but low power consumption during communication and event processing. Our architecture is designed to cover all subsequently mentioned requirements. The whole system is divided into two layers, the system and the application layer, see Fig. 2.

A. Architectural Overview

The system layer contains the AVS-Extrem Sensor Board, the operating system and the energy management. The AVS-Extrem Sensor Board is an ARM7 based wireless sensor node which is designed to fit into typical construction site fence elements and will be introduced in Section III-B1. A low power acceleration sensor is needed to gather data of any fence movement and is evaluated in [8], where the SMB380 Bosch sensor is suggested as a suitable 3D acceleration sensor for the fence monitoring application. The system has to be applicable for rough environments, severe weather conditions and strong shocks. We developed a specialized and waterproof housing that will be described in detail in Section III-B2. A flexible energy management that supports WOR and numerous MCU dependent energy saving techniques are required. The energy supply has to last for a period longer than three months of use, to meet the minimum requirements of a short term installation. The energy management is described in Section IV-A and Section IV-B. To guarantee a low response time and high precision during the event detection, a thread based real time operating system (RTOS) is needed. We use the FireKernel RTOS [9] which features threading and a priority based preemptive multitasking. For completely secured communications, we implemented a security layer that proposes a symmetric cipher-algorithm in CBC-Mode [10] to provide confidentiality. A message authentication code (MAC) is calculated over the packet which includes the encrypted

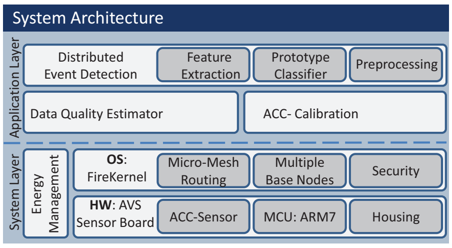

Fig. 2. System architecture

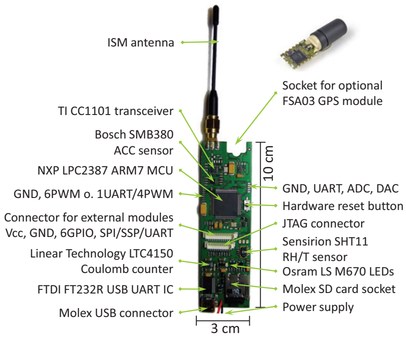

Fig. 3. AVS-Extrem Sensor Node

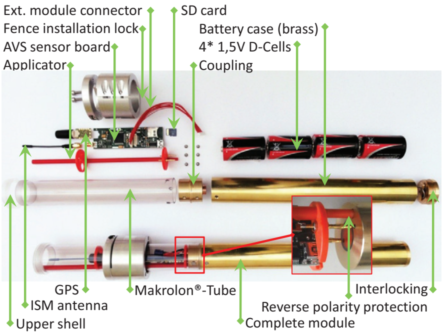

Fig. 4. AVS-Extrem housing

payload, to guarantee authenticity and integrity. Each packet header contains a counter to provide a simple replay protection. Initialization vector based encryption of the payload and the MAC calculation ensures semantic security. Further details are represented in [11]. A self healing sensor network with dynamic routing covers changing communication characteristics. The Micro Mesh Routing (MMR) protocol, initially introduced in [12].MMR is briefly described in Section III-C. Furthermore, we extend MMR by a virtual sink to support multiple base nodes. This enhances our WSN with a handy scalability that depends on the number of base nodes. Additional nodes can be added with minimal effort.

The application layer contains an optional Dempster-Shafer based data quality estimator [13]. The data quality estimator is able to assess incoming values by heuristics. A data quality estimation allows subsequently to decide whether the reliability of a measurement is high or low. The data aggregation can be performed depending on the data quality. The acceleration sensor needs to be automatically calibrated before and after events. We developed our own application dependent calibration routine that takes into account noise and interruptions during the calibration period. The system also contains the mandatory distributed event detection system that is described in Section III-D, while further details can be found in [4].

B. Hardware

The AVS-Extrem sensor node is based on the MSB-A2 platform [14] developed at the *Freie Universität Berlin*. As like the MSB-A2, the AVS-Extrem sensor node is equipped with an LPC2387 microcontroller [15] and uses the CC1101 transceiver [16]. The microcontroller is based on an ARM7 core, operating at $72\,\mathrm{MHz}$. The CC1101 is driven by a $26\,\mathrm{MHz}$ clock and uses the $868\,\mathrm{MHz}$ SRD radio band. Because of our requirement of a revised blank shape and additional peripheral devices, we designed a new circuit, see Fig. 3.

1) AVS-Extrem Sensor Node: For the AVS-Extrem project a new printed circuit board (PCB) had to be designed due to the fact that an accelerometer is required for our application and the blank shape of the PCB had to be adjusted to fit into the proposed housing. The core of the new design is identical to the MSB-A2, but new peripheral parts were added.

The 3D-accelerometer SMB380 from Bosch [17] is applied because of its supplementary features. The autarkic acceleration sensor logic (ACC-Logic)is able to detect the event beginning with a configurable threshold register, here called *ACC-Logic*. During this detection it is important to mention that the MCU is in power down mode (PD). The sensor offers a high sampling rate of up to $1500\,\mathrm{Hz}$ and a maximum sensitivity of $8\,\mathrm{g}$ which is necessary to cover high acceleration like shocks at the fence. As the detection system is not allowed to switch off the acceleration sensor, a low power consumption of $200\,\mu\mathrm{A}$ at $3.3\,\mathrm{V}$ is inevitable. The sensor raises an external interrupt to wake up the system for further processing once a certain acceleration force above a user defined threshold occurs.

In addition to the core components explained above, the PCB also uses a temperature/humidity sensor to monitor environmental parameters as well as a SD Card slot to store prototype data. The LTC4150 coulomb counter is used to monitor the state of the batteries. Further peripheral interfaces depicted in Fig. 3 are used for development purposes.

2) Sensor Node Housing: Our aim is to integrate as much of the sensor node as possible into the vertical rod of a zincked steel fence, because the fence itself already offers a high grade of robustness, see Fig. 5. In contrast to this benefit, the antenna has to stay outside the fence as the vertical rod affects wireless communication like a Faraday cage. The housing enables to arrange experiments with comparable settings as the sensor nodes can be mounted in a highly repeatable, very stable and fixed way within the fence. To integrate the sensor node, the housing and the power supply into the fence, we solve several engineering problems that typically are overseen or ignored in testbed deployments, see Fig. 4. Our primary design-goal is to integrate the AVS sensor node and standard D-Cells into the housing. We decided to use standard D-Cells to provide a wide design and to deliver a more stable capacity even at low temperatures. We further decided to choose a brass battery case for solidity and conductivity for the ground connection. A brass coupling connects the sensor node with the battery case while a thin rigid PVC isolated copper core connects the positive pole. To provide undisturbed radio communication and to assure resistance to a wide range of weather conditions, we package the sensor node within a Makrolon® tube. The red Makrolon® applicator (see Fig. 4) holds the sensor node

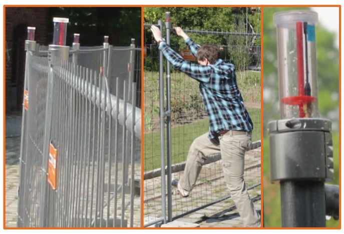

Fig. 5. Experiment at the fence & integration of sensor node into vertical fence rod

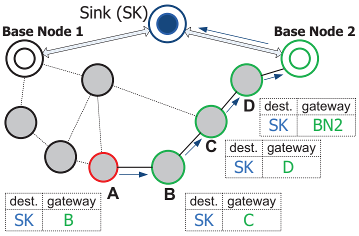

Fig. 6. Data transmission over base nodes to sink by using MMR

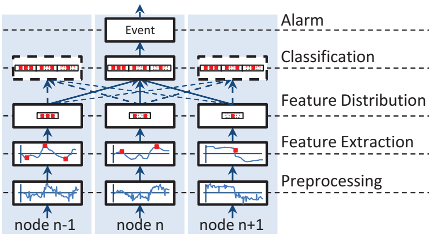

Fig. 7. Distributed Event Detection: Data flow from event to alarm

in correct position, is plugged into the coupling and employs standard banana plugs. A reverse polarity protector allows only one correct contact termination. The battery case is shuttered with a screw cap which is actuated by a spring. In rare cases a vertical shock is able to compress the spring which disconnects the battery. This problem is solved by an additional capacitor to bypass a temporary interrupt to the power supply. To affix the sensor node in the fence a two part installation lock is used to fix the node at the lock and the lock at the fence.

C. *Networking – Multiple Base Nodes*

Multiple base nodes are adopted in order to provide fault-tolerant connections between an arbitrary node in the WSN and the virtual sink. Fig. 6 illustrates a configuration of a network with two base nodes. Base nodes act as proxies between the sensor nodes and the sink. The sink is connected to multiple base stations via a wireless or wired connection.

In case of failure or non-reachability of a base node, an alternative base node is automatically discovered by the Micro Mesh Routing Protocol (MMR) [12]. Thus, the reliability of the network and the accessibility of the sink will be improved. The Micro Mesh Routing Protocol as a reactive and dynamic routing protocol selects a robust path to the sink. MMR combines different approaches and features of divers routing protocols like AODV (Ad hoc On-Demand Vector) to use hop-by-hop routing during the data transmission, while the principle of DSR (Dynamic Source Routing) is used to collect partial route information in the course of the route discovery phase. MMR attempts to set up a stable route between two neighbors. This neighbor knowledge accomplishes efficient power consumption during the communication with the sink.

In contrast to a strict reactive protocol, the intermediate nodes analyze all forwarded packets, to update their routing tables. Sequence numbers are used to prevent loops, to avoid duplicates and to determine the freshness of a route. MMR handles changes in the WSN topology by automatically discovering new routes to the base nodes. The network address of the sink is known by the base nodes and used to forward packets directly to the sink.

If a node intends to send a packet to the sink, without having an appropriate routing table entry, a route discovery will be started. The discovery operation uses two message types *Route Request* (RREQ) and *Route Reply* (RREP). The base nodes handle all RREQ messages destined to the sink by generating a RREP message, hence, each base node virtually represents the sink. After route discovery, the data transfer can be started, where the intermediate nodes forward the packet to one of the base nodes according to the routing table entries. Finally, the base node delivers the message to the sink, see Fig. 6.

D. *Distributed Event Detection*

The event detection system consists of two modules: *training* and *recognition*. Both modules perform a collection of raw data during an event, preprocessing, feature extraction, feature distribution and a final classification which may raise an alarm in the recognition module, see Fig. 7.

1) Training: During the a priori and supervised training all events have to be trained at the fence where the events need to be performed as exemplary depicted in Fig. 5. The measured raw data is preprocessed with filter functions to smoothen the incoming values. During preprocessing, the system additionally segments the event by hysteresis functions to detect the beginning and the end of events. Depending on the application and the used features it is subsequently possible to normalize the data by value and/or by time. During the feature extraction, the nodes calculate all available features for each trained class and send the features in a combined feature vector to the sink. The feature generation reduces the dimensionality of the data stream but keeps the characteristics of varying events. The Leave-one-out Cross Validation (LOOCV) [18] is applied on all collected features to assess feature combinations. The assessment is done by the Euclidean-based prototype classifier.

The selected features are combined in a new reference feature vector for each class: called *prototype*. The variance in the raw data during training is used to define a radius around each prototype, the classification region. Events classified within this region are assessed as valid. The training is finished after distributing the prototypes to all sensor nodes.

2) Event Detection: If an event arises at the fence the affected nodes start to collect raw data, similar to the training. First of all, the raw data is preprocessed and in contrast to the training only the selected features, defined by the prototypes, are extracted. During feature distribution, all features are send via broadcast to the 1-hop neighborhood. Nodes affected by the event fuse the received features to a complete but initially unclassified feature vector. The nodes then run the euclidean based prototype classification. If a node classifies an event successfully, the event can raise an alarm. As in [4] suggested, only one small alarm packet has to be send to the sink in.

IV. Energy-Awareness

A. Hardware

The main goal of our new hardware platform – next to meeting size limitations, offering sufficient performance and providing required resources – is a long runtime between maintenance intervals to increase usability. We approach these requirements by choosing components that match to the expected duty cycle of most WSNs: Long term inactivity and short term processing and communication. Optimizing this period of inactivity obtains higher energy saving and increases the lifespan many times. Nevertheless, our primary requirement is a continuous monitoring of occurring events. We fulfill this requirement by a configurable 3D-acceleration sensor that is able to activate the MCU for further acceleration data processing after an event occurred. During surveillance, the system has a total power consumption of $9.0\,\mathrm{mW}$ and about $0.7\,\mathrm{mW}$ are attributed to the acceleration sensor. The remaining power consumption is composed by the ARM7 power down mode, temperature/humidity sensor, transceiver WOR-mode and unavoidable transformation loss. In contrast, event detection without ACC-Logic would require at least $206.25\,\mathrm{mW}$.

B. Software

The FireKernel [9] implements a generic support for power savings. It enters power down mode whenever threads do not demand processing time and no peripherals are operating that will be affected by power down mode. Peripherals are always affected whenever their operation is dependent on clock signals generated indirectly by the Phase-Locked-Loop. The device driver needs to prevent FireKernel from accessing the MCU power down mode in these cases. Nevertheless, the MCU can be halted by using IDLE mode.

Furthermore, whenever no fine-grained timers are required but a rough time control is sufficient, the application can use Real Time Clock (RTC) alerts with a one sec resolution. The MCU can be reactivated by the RTC alert which enables that all other clocks can be halted during idle time. The external RTC $32\,\mathrm{kHz}$ clock will not be affected by this power down mode and continues to operate normally and alerts the MCU just in time.

V. Experiments & Results

Our experiments show a whole life cycle of a sensor node during distributed event detection and recognition. To measure the energy consumption of the whole board as accurately as possible we soldered a $10\,\Omega$ shunt resistor into the supply line which is powered by a reference voltage of $5\,\mathrm{V}$. To measure the voltage of the shunt resistor a digital sampling oscilloscope (DSO) is attached. As the resistor and the voltage are known, we can calculate the value of the current and use it to calculate the electric power used by the sensor node over the time of one DSO sample. By integrating the electric power over the time of one system state, like packet transmission or IDLE mode, we can exactly measure the energy used per state and use this information to approximate the energy consumption of the whole system over a certain time.

During the event detection phase, the sensor nodes use the MCU power down mode (PD), that also shuts down all internal peripherals. The wireless transceiver uses the WOR mode and is able to process incoming data. The acceleration sensor is active and monitors the fence elements movement. Fig. 10 demonstrates the relation between different energy-aware techniques that can be utilized with our hardware. An energy diagram of all phases of an event is depicted in Fig. 8.

During PD, a mean energy consumption of $9.0\,\mathrm{mW}$ is ascertained. During an event, the MCU is periodically utilized to fetch acceleration data from the acceleration sensor to the MCU ($206.25\,\mathrm{mW}$). This is followed by the feature extraction ($350\,\mathrm{mW}$), and classification ($58.80\,\mathrm{mW}$ on average). As described in [4], a maximum of seven sensor nodes are involved into a fence event. Hence, in the phase of feature distribution one broadcast packet is send ($373.33\,\mathrm{mW}$) and during classification six packets are received ($178.5\,\mathrm{mW}$) from the neighborhood. Finally the result is calculated and sent to the sink. Afterwards the sensor node is re-calibrated or if the hysteresis function has converged, the sensor node immediately returns to the detection mode. The average time duration of an event is $10\,\mathrm{s}$, including sampling, feature extraction, distribution and classification is $129.8\,\mathrm{mW}$.

To evaluate radio communication latency we deployed 28 AVS-Extrem sensor nodes indoors (in the university building) and outdoors. We measured the latency for up to three hops and compare WOR with CRX. Latency is here defined as the roundtrip time of a full-sized data-packet.

Fig. 9 shows the results of the latency experiment in multiple settings. The CRX results represent the latency values with the transceiver in constant RX mode. In CRX mode we assume that the transceiver causes no delay. During WOR we expect higher latencies because of a duty cycle of 0,0018% ($542\,\mathrm{ms}$ sleep period, $1\,\mathrm{ms}$ awake period). For WOR mode, we reach a median indoor latency of $633, 1141, 1611\,\mathrm{ms}$ and outdoor latency $766, 1313, 1893\,\mathrm{ms}$ for 1 to 3-Hop routes. CRX-environments reach $62, 192, 324\,\mathrm{ms}$ respectively.

VI. Evaluation

Due to energy-aware techniques (EAT), we are able to reduce the mean power consumption from $456\,\mathrm{mW}$ to $9\,\mathrm{mW}$, see Fig. 10. To evaluate energy consumption in a real-world scenario we assume 5 events per hour with a sampling duration of $10\,\mathrm{s}$ each. Additionally 3 unicast packets are assumed for status packets. As construction site fences are rearranged infrequently, the routes are assumed as stable. The distributed event detection requires an additional power of $4\,\mathrm{mW}$, while it requires additional $197\,\mathrm{mW}$ if the ACC-Logic is not active.

Hence, we measured a mean power consumption of $13.0\,\mathrm{mW}$ in the detection scenario. Four standard D-Cells are capable of $90\,\mathrm{Wh}$ resulting in a lifetime of about 280 days, as depicted in Fig. 10. This outperforms the fence surveillance system of Kim et al. [1] by four times, while they assume that no event will occur. Even in case of CRX or very high

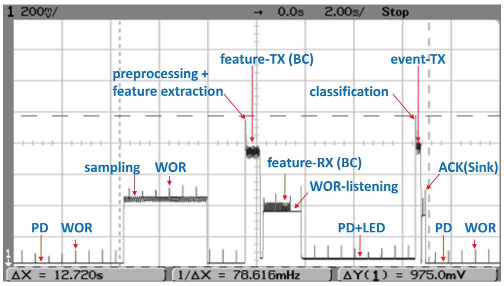

Fig. 8. Energy diagram for event processing

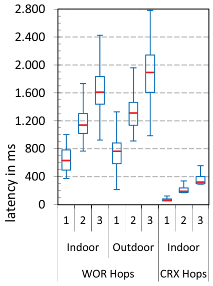

Fig. 9. WOR- and CRX-Latency

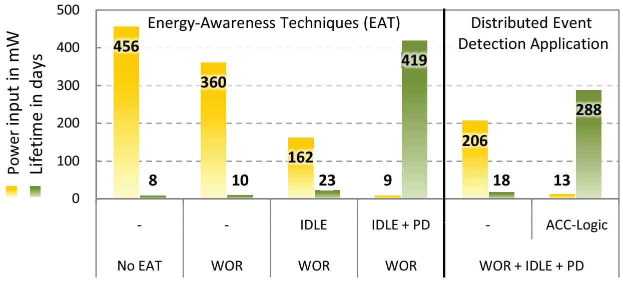

Fig. 10. Energy consumption and lifetime

frequency of events, our system will survive of at least one week which serves short term installations, see first column (No EA) in Fig. 10.

Placing the initial detection of an event into the sensing hardware by applying the previous mentioned ACC-Logic, we gain energy savings in an order of 15 times compared to a software detection (no ACC-Logic), see Fig. 10. This high impact emphasizes the importance of a MCU independent sensor. The ACC-Logic decreases MCU processing time and enables the full capabilities of energy saving modes.

WOR latency in Fig. 9 can be verified by summing up the following values: As we use a WOR sleep period of $542\,\mathrm{ms}$ we expect a delay of about $280\,\mathrm{ms}$ per transmission. Hence, as we measure the latency by using a roundtrip packet we assume about $560\,\mathrm{ms}$ per hop. In addition a random CRX-backoff, discussed in [12] and internal computation time occurs, which can be approximated from the CRX 1-hop time of about $60-100\,\mathrm{ms}$. The results show an expected latency of $633-766\,\mathrm{ms}$ for a 1-hop WOR echo transmission up to $1611-1893\,\mathrm{ms}$ for a 3-hop WOR echo transmission. The outdoor scenario results in higher latencies due a higher inter node distance compared to the indoor scenario. The outliers of the measurement can be explained by packet loss effects caused by bad link quality. Each additional hop adds approximately $100\,\mathrm{ms}$ delay to the transmission for CRX and $500\,\mathrm{ms}$ for WOR.

VII. Conclusion

We presented a comprehensive system for fence surveillance by taking real-world requirements that cover practical scalability by employing multiple base nodes and energy-awareness in a wireless sensor network into account. Further we obtain a ubiquitous integration of the modular sensor node with its power supply and housing into a fence. To outreach these results we developed a robust and weather-proof housing that contains a replaceable power supply for expeditious sensor node deployment and maintenance. Our expected lifetime of more than 280 days outperforms the current state of the art by four times with a lifetime. Next to our extended support of power saving functions in hardware and software, lifetime could be extended thanks to the energy-awareness of WOR which has been utilized in our system. The expense of increased latency is acceptable and fulfills the requirements in our scenario to provide a responsive distributed fence surveillance system.

Acknowledgment

This work was funded in part by the German Federal Ministry of Education and Research (BMBF, AVS-Extrem).

References

[1] Y. Kim, J. Kang, D. Kim, E. Kim, P. Chong, and S. Seo, "Design of a fence surveillance system based on wireless sensor networks," in *Autonomics*, A. Manzalini, Ed., 2008, p. 4.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification (2nd edition)," 2000.

[3] I. F. Akyildiz and M. C. Vuran, *Factors Influencing WSN Design*. John Wiley and Sons, Ltd, 2010.

[4] G. Wittenburg, N. Dziengel, C. Wartenburger, and J. Schiller, "A System for Distributed Event Detection in Wireless Sensor Networks," in *Proc. of 9th ACM/IEEE Intl. Conf. on IPSN*, Stockholm, Sweden, 2010.

[5] A. Yousefi, S. Member, A. Dibazar, and T. Berger, "Intelligent fence intrusion detection system: detection of intentional fence breaching and recognition of fence climbing," in *Technologies for Homeland Security, 2008 IEEE Conference on*, 2008, pp. 620 –625.

[6] *TEDAS^{TM}- Sensor Network System for Perimeter Security*, Genet©, ISTANBUL /TURKEY.

[7] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, pp. 10:1–10:29, 2009.

[8] N. Dziengel, M. Ziegert, Z. Kasmi, F. Hermans, S. Adler, G. Wittenburg, and J. Schiller, "A Platform for Distributed Event Detection in Wireless Sensor Networks," in *Proc. of the 1st Intl. Workshop on Networks of Cooperating Objects (CONET '10)*, Stockholm, Sweden, 2010.

[9] H. Will, K. Schleiser, and J. Schiller, "A real-time kernel for wireless sensor networks employed in rescue scenarios," in *Proc. of the 34th IEEE Conference on Local Computer Networks (LCN)*, New York, Piscataway, USA, 2009, pp. 834–841.

[10] "NIST, National Institute of Standards and Technology, DES Modes of operation." Federal Information Processing Standards Publication 81 [FIPS 81], 1980.

[11] N. Schmittberger, "Security in Event-Driven Wireless Sensor Networks," Master's thesis, Department of Mathematics and Computer Science, Freie Universität Berlin, 2011.

[12] T. Hillebrandt, "Untersuchung und Simulation des Zeit- und Energieverhaltens eines MSB430-H Sensornetzwerkes," Master's thesis, Freie Universität Berlin, 2007.

[13] F. Hermans, N. Dziengel, and J. H. Schiller, "Quality estimation based data fusion in wireless sensor networks," in *MASS*, 2009, pp. 1068–1070.

[14] M. Baar, H. Will, B. Blywis, T. Hillebrandt, A. Liers, G. Wittenburg, and J. Schiller, "The ScatterWeb MSB-A2 Platform for Wireless Sensor Networks," Freie Universität Berlin, Berlin, Germany, Tech. Rep., 2008.

[15] "LPC2387 datasheet," NXP, Eindhoven, Netherlands. [Online]. Available: "http://www.nxp.com/documents/datasheet/LPC2387.pdf"

[16] "CC1101 datasheet," Texas Instruments, Texas, USA. [Online]. Available: "http://focus.ti.com/lit/ds/swrs061f/swrs061f.pdf"

[17] "SMB380 datasheet," Bosch, Reutlingen, Germany. [Online]. Available: "http://www.bosch.de/start/content/language2/downloads/news_0505_1_Sensortec_Datasheet_en.pdf"

[18] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. of the 14th int. joint conf. on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995.