Field experiments with Wireless Sensor Networks

Joakim Flathagen, Reinert Korsnes, Terje M. Mjelde, Vinh Q. Pham and Jostein Sander
Field experiments with Wireless Sensor Networks

Joakim Flathagen, Reinert Korsnes, Terje M. Mjelde, Vinh Q. Pham and Jostein Sander

Norwegian Defence Research Establishment (FFI)

18 June 2014
Emneord

Styrkebeskyttelse

Sensorer

Sensorintegrasjon

Perimeterbeskyttelse

Trådløse sensornettverk

Godkjent av

Torunn Øvreås  Project Manager

Anders Eggen  Director
English summary

The motivation behind the work in this report is to contribute to improved force protection by instrumenting the perimeter of the base with a wireless sensor network. We report on experiences with the implementation, deployment, and operation of a surveillance wireless sensor network. We have developed a Situational Awareness Sensor Network (SASS), consisting of 50 TelosB based nodes with infrared, radar and acoustic sensors. The report presents several experiments with the sensor network deployed in a military training facility. The results from the experiments include the network performance of the sensor network as well as a quantification of the operative effect of the system. Finally, we share our experience in building sensor nodes and lessons learned in conducting efficient field system trials.
Sammendrag

## Contents

1. **Introduction** 7  
2. **Related Work** 8  
3. **System goals and design considerations** 9  
4. **SASS: Situational Awareness Sensor System** 10  
   4.1 Architecture 10  
   4.2 SASSNode 11  
   4.3 Sensing 11  
   4.4 Networking 13  
   4.5 Time Synchronization 14  
   4.6 Localization 14  
   4.7 Software Considerations 15  
5. **Field Test Experiments - Perimeter Surveillance** 16  
   5.1 Experimental setup 16  
   5.2 Results 17  
6. **Field test experiments - Road Surveillance** 18  
   6.1 Experimental setup 18  
   6.2 Results 18  
7. **Network performance** 19  
8. **Lessons Learned** 21  
9. **Conclusion** 22
1 Introduction

Military base protection covers all protecting and safeguarding activities such as counterintelligence, physical security and surveillance. It is a very difficult task to offer adequate protection against pertinent threats both for permanent bases and base camps. Improving the force protection is therefore an everlasting and multi-disciplinary endeavor involving the military, the industry and academia. The motivation behind this report is to contribute to improved force protection by instrumenting the perimeter of the base or a nearby area with a Wireless Sensor Network (WSN). With timely sensor data from discrete and autonomous sensors, the warfighters are provided with early warning and greatly enhanced situational awareness. By speeding up the chain of detection, identification and neutralization, the level of protection is enhanced while conserving available human resources.

The idea of instrumenting the tactical domain with wireless sensors dates back to the late sixties, realized by the Air Delivered Seismic Intrusion Detector system by the US Air Force in the Vietnam war [1]. Even if the first systems for autonomous and wireless surveillance were cumbersome, heavy, and with unpredictable operative behavior, they paved the way for future research. The proliferation of Internet technology and miniaturized electronics has brought forward the vision of Wireless sensor networks. Several such networks now exist for various applications within structural monitoring, volcano monitoring, sheep tracking, to name a few. Although wireless sensor network research is just over a decade old, there is strong evidence that WSNs can offer a new paradigm for tactical surveillance. The small physical appearance makes the sensor devices stealthy in a hostile environment and less subject to vandalism or theft compared to traditional tactical sensor systems. Furthermore, the network protocol redundancy and the vast number of sensing nodes can improve reliability and minimize the false alarm probability.

![Diagram of perimeter security and road surveillance](image)

Figure 1.1  The report focuses on using Wireless Sensor Networks in two base protection scenarios: Perimeter security and road surveillance.

In this report we describe our experiences with the implementation, deployment, and operation of a wireless sensor network for military base protection. For the purpose of this report, we developed a Situational Awareness Sensor System (SASS) consisting of 50 wireless TelosB based nodes equipped
with radar detectors, microphones and passive IR sensors. The sensor nodes used multihop routing to report alarms and network monitoring information back to the headquarter. We conducted a series of experiments in-door and out-door to test the different software and hardware components. The experiments included dozens of different networks of 3-50 nodes, from indoor office testbeds, urban areas, to full-scale tactical field tests. In the main military base protection experiment, the nodes were deployed in a military tactical training area for testing and evaluation in a realistic setting. The experimental results we obtained show that the operative effect of using a sensor network for perimeter surveillance is potentially very high: WSN systems in this category contribute to improved situational awareness, conservation of human resources, and at the end of the day, they can spare life. The results also yield many insights relevant for sensor network applications outside our particular research interest. Our measurements of packet loss and network scalability provide a deeper understanding of how WSNs perform in real-life. We also share our lessons learned with designing and deploying the sensing nodes.

The rest of this report is organized as follows. Section 2 reviews related work. Section 3 describes the system goals of a typical force protection surveillance WSN. Section 4 describes our sensor system. The field test experiments are described in section 5 and section 6, while section 7 reports on the network performance. After that, section 8 summarizes the lessons learned while section 9 concludes the report and proposes directions for future work.

2 Related Work

The Great Duck Island project is considered one of the first known deployments of a WSN [2]. The purpose of the project was to deploy 32 sensor nodes for monitoring nesting burrows. Unfortunately, the system had very high node failure rates, but the authors gave insight about several deployment issues such as the importance in providing self-maintenance of the sensor network and to correctly packing the sensor nodes. The potato-field project from Delft University in [3], is another example of a deployed network. The project clearly demonstrates that performing a real-world experiment is far from a trivial task. Although their results did not meet the expectations from an agricultural point of view, they gave good insight in the typical problems overlooked when WSNs are brought to the field. The work of Barrenetexa et al. [4] also serves as an excellent guidance summing up the potential problems deploying sensor networks.

The common denominator in most of the deployed sensor networks in the literature is that the typical end-users targeted are researchers. Typical research-oriented applications are for example habitat monitoring [2], agricultural monitoring [3], environmental monitoring [4] or volcano monitoring [5]. These WSNs are built by and for researchers interested in collecting certain measurement data from the physical world for subsequent analysis. A military base protection scenario on the contrary, targets a warfighter end-user. Compared to researchers, which studies sensor data for a certain research project, a warfighter must take quick operative decisions based on the information generated by the WSN. This puts extremely high requirements to the reliability and ease-of-use of the system. Due to the harsh environmental and operative requirements in the military scene, existing commercial
Unattended Ground Sensor systems (UGS), has limited lifetime and substantial weight and size. Hence, they generally do not yet fulfill the vision of the WSN-concept. One of the earliest examples of a deployed wireless sensor network for tactical purposes is the work by Arora et al. [6]. They present an experimental system for intrusion detection and target classification using 90 wireless sensing nodes equipped with radar and magnetic sensors. VigilNet [7] is another WSN surveillance system comprising 70 sensing nodes with magnetic sensors. By employing alarm aggregation among sensing nodes, the system shows low probability of reporting false alarms. Finally, Rothenpleier et al. have presented FlegSens [8], an experimental surveillance WSN based on passive infrared sensors (PIR) for detecting trespassers. Their prototype implementation consists of 16 nodes. Our system is inspired by the related works, but distinguishes from them in the combination of sensors used and the specific scenario we attack.

3 System goals and design considerations

The operational problem motivating the work in this report is military protection by the means of unattended ground sensors. The objective of such an application is to provide an early warning system alerting the military command and control unit about the presence of enemy personnel or vehicles. Specifically we are interested in two surveillance scenarios (cf. Fig. 1.1). The first scenario we target involves monitoring of the perimeter around a base. This involves the detection and tracking of personnel (adversaries) entering the perimeter. The objective of the second scenario is to monitor abnormal and suspicious behavior along roads. The deployment of Improvised Explosive Devices (IEDs) is one type of such abnormal behavior that we wish to detect and identify.

The first scenario, the perimeter scenario, requires the detection of personnel moving in a forestall environment. A challenge here is that the movement of the personnel can be very low and that the forest can be very dense. In such a perimeter scenario we expect very few (if any) alarm messages during the entire operational lifetime of the sensor network. Further, the network is in this case dense (limited to a relatively small area) but with very few line-of-sight links due to trees, bushes and other obstacles.

The road scenario in contrast, focuses on the detection of vehicles moving at velocities up to 25m/s. Although individual alarms are filtered based on the behavior of the vehicle, we can expect up to several alarms per minute per sensor depending on the traffic along the road. Further, the network is sparse and stretched over a long area, but has mostly line-of-sight links. Both scenarios dictate the design of the sensor devices we produce. Thus, we need a flexible platform for the experiments. We briefly sum up the application requirements that must be satisfied in the following:

Sensing: The sensors must be able to detect the presence of both vehicles and personnel. In order to accommodate the different scenarios specified, the sensitivity and the function of the sensors must be tunable. There is a correlation between the ability to detect all possible targets and the probability of detecting false events. Depending on the mission, and the tolerance of false alarms, the sensors could be subject to fine-tuning both prior to and during the mission.
Networking: The limited transmission range of typical sensor nodes is not sufficient to guarantee full coverage in the monitored area. This implies that there is a need for a routing protocol to transmit messages via intermediate nodes. The networking subsystem must also handle the bursts of messages that can occur upon intrusion. But at the same time, the network load must be at a minimum when the sensor network is idle, both to reduce power dissipation and to minimize RF interception. Since the sensing nodes are deployed in large quantities in an environment which is difficult to access, there is a need to remote-control the sensors from the base-station. Hence, the system requires protocols enabling multihop communication in two directions (sink-to-base and base-to-sink).

Longevity: A surveillance application for force protection has a typical expected lifetime from a few days to several months. Due to the hostile nature and the vast number of sensing nodes it might be impractical or impossible to manually change batteries during the operation. Hence, there is a need for systems that are energy-aware. However, since we are dealing with a test system, we do not specify a minimum system lifetime at this stage.

Deployment and usability: Stealthiness is important in most tactical operations. One of the major benefits of small sensing devices is that their physical appearance makes the devices stealthy in a hostile environment and less subject to vandalism or theft compared to traditional tactical sensor systems. Since the sensors are used in a tactical setting, they could be deployed any time during the day and in different weather conditions, they must be simple to use and handle.

Since the system was meant to work in an operational setting, our prime focus was on risk reduction. This made us trade off general or optimal solutions for simple ones. The next section provides the detailed implementation of our sensor system and explains the choice of solutions and protocols.

4 SASS: Situational Awareness Sensor System

4.1 Architecture

Based on the goals and design considerations discussed above, we developed a system for increased situational awareness for force protection: Situational Awareness Sensor System (SASS). The architecture, shown in Fig. 4.1, consists of (from left to right in the figure) a localization component, the network of wireless sensors (SASSNodes), a sink node collecting alarms connected to an operator machine (Controller). The Controller establishes and configures a surveillance session. In addition we developed a trial analysis tool with a mapping software. Further, it is possible to present sensed data and alarms to a Battlefield Management System (BMS). Our previous work show how this can be done [9].

In the following, we present the development of the SASSNode and the sensors. Furthermore, we describe the networking system and the different system components.
Our Situational Awareness Sensor Node (SASSNode) is based on the TelosB mote [10], a typical WSN device (cf. Fig. 4.2a). The TelosB has a 4 MHz MSP430 processor, 10 KB of RAM and 48KB program memory (flash). TelosB uses the Chipcon CC2420 radio in the 2.4GHz band, an IEEE 802.15.4 compatible radio at 250kbps. The sensor node runs the TinyOS 2.x operating system, which is a flexible and modular operating system designed for low-power sensing devices. The node is equipped with sensors and batteries and is mounted in a sensor enclosure of 10x12x12cm (shown in Fig. 4.2d). The enclosure was designed with simplicity in mind and the time limitation of the project restricted us from building sensor nodes with full weather proofing.

4.3 Sensing

Each SASSnode is equipped with three sensors: A passive IR detector (PIR), microphone and a Doppler radar. The PIR and the microphone are mounted on a custom sensor board (cf. Fig. 4.2b). The PIR detector provides excellent sensitivity and has low energy consumption and signature due to its passive operation. The main weakness with PIR sensors is their sensitivity to sunlight and objects emitting IR, making them prone to producing false alarms. The microphone gives acoustic detection of targets and is most relevant when detecting objects with distinct sound emission (e.g., vehicles). The microphone was used only in the road-surveillance experiments described in this report. The third sensor is a BumbleBee Pulsed Doppler Radar (cf. Fig. 4.2c). The radar works at 5.8GHz and has a maximum detection range of 10m. The radar is an excellent supplement to PIR, since it produces no false alarms due to sunlight. On the other hand, it is sensitive to wind blown grass or trees. It also has an active nature, consuming considerable more energy than the PIR detector.

The sensing algorithms for IR and the radar are similar. Both sensors are connected to the AD converter, which samples the sensors at 50Hz frequency. The algorithm is simple: Let $m$ denote the set of sensor samples and $m(k)$ the last sample. The alarm decision ($a$) is based on counting the number of collected samples $N_a$ that is above the alarm threshold target $tt$ within a threshold period.
Figure 4.2 The SASS sensor node (SASSnode) consists of (a) TelosB mote, (b) custom sensor board with PIR and microphone, (c) a Bumblebee Doppler radar circuit board and (d) a casing, here fully assembled. Photos by FFI.

If the number $N_s$ is higher than the trigger threshold $a_{tt}$ an alarm is created. More precisely:

$$a = \begin{cases} 
1, & \text{if } N_s > a_{tt}; \\
0, & \text{otherwise} 
\end{cases} \quad (4.1)$$

where

$$N_s = \sum_{i=0}^{a_{tp}} \begin{cases} 
1, & \text{if } |m(k+i)| > tt; \\
0, & \text{otherwise} 
\end{cases} \quad (4.2)$$

To account for background noise, the threshold target $tt$ adapts according to the signal value. Fig. 4.3a and Fig. 4.3b show the signal value (black line) from the PIR and the radar sensor respectively. The green area defines the given threshold while the red areas defines the time periods when the signal $(m(k+i))$ exceeds the threshold. Both figures show a silent period (i.e., no detection) followed by a detection period initiated by a human entering the detection area.

The purpose of the sensing algorithm for the microphone is to detect vehicle activity. It samples data at 4kHz and uses a simple approach: If the input signal is above a threshold $A_t$ for more than a defined period $A_p$, the data series is stored. Subsequently, the symmetrical property of the series is evaluated to find the exact time of the vehicle crossing.

With both PIR and radar, it is difficult to meet the low false alarm rate requirement and still have the sufficient sensitivity. Since both sensors have a significant probability of falsely reporting events that
Figure 4.3 The black lines in (a) and (b) show the raw output \( (m) \) from the PIR and radar sensors respectively. The green areas show the threshold \( (tt) \) while the red areas illustrate the alarm samples (i.e., \( m > tt \)).

are not actually present, some sort of data aggregation or filtering is necessary. Such functionality is usually implemented by letting designated network nodes wait for multiple reports, either from the same node generating the first observation (temporal redundancy), or from neighbors of this node (spatial redundancy), before creating the final alarm. This can be implemented either at the central sink, or it can be performed distributed in the network. Although the latter category (i.e., in-network data aggregation) is more efficient in terms of packet transmissions, it is severely more complex to implement than relying on data aggregation only at the sink end. For the experiments in this report, we rely on aggregation at the sink (i.e., the Controller software). The main reason for this, is that this approach enables us to collect all sensor readings (unfiltered) for later analysis.

4.4 Networking

A plethora of routing protocols for WSNs has been developed within the last decade. We initially considered four different protocols for the use in our force-protection scenario: TYMO [11], Multi-hopLQI [12], CTP [13] and a simple broadcast protocol. TYMO originates from the ideas behind AODV [14], which is a protocol tailored to mobile ad-hoc networks. In TYMO, the routes are based on the Point-to-point (P2P) principle meaning that any two nodes in the network can establish routes and communicate. MultihopLQI and CTP on the other hand, are Multipoint-to-point (MP2P) protocols. Here, the sink node constitutes the root in the routing tree. MultihopLQI uses the Link Quality Indicator (LQI) from the physical layer to additively obtain the path towards the sink, while in CTP the Expected Number of Transmissions (ETX) is used as the metric in the routing decision. Starting with an ETX of 0 at the sink, each node calculates its own ETX as the ETX reported by the parent plus the ETX of the link to the parent. Finally, we implemented a simple and naive broadcast protocol (BCAST). Our BCAST implementation works as follows: Message originators send all packets as broadcast packets. A node hearing a BCAST transmission, records the sequence number and the originator (to avoid duplicate retransmissions) and retransmits the packet. Eventually, the packet reaches its destination (i.e., the sink).

To evaluate the performance of the four routing protocols in a realistic setting, we implemented an indoor testbed consisting of 20 TelosB-based SASSNodes. The testbed was configured as a multihop
network with an average node degree of 6. The packet rate was fixed at one packet per node per 20s and we ran 10 experiments lasting one hour for each protocol. Fig. 4.4 shows the packet delivery ratio for each routing protocol. The figure also shows the average number of data packets transmitted per node during the test. The results show that CTP gives the best balance between packet delivery ratio and overhead. Based on these results, we decided to use CTP as the routing protocol for our system. We refer to [15] for further results based on a broader set of scenarios.

CTP is incapable of Point-to-multipoint communication (P2MP). Hence, to deliver information to all nodes in the WSN a dissemination protocol is required. Since messages can be lost due to e.g., collisions, channel noise or even buffer overflow, the dissemination protocol needs to be reliable. In addition, message synchronization could be necessary after a node reboot, e.g., if an application failure causes the watchdog timer to elapse. This means that simple flooding of the queries is not sufficient. In our implementation we have used Drip dissemination [16] to account for message synchronization. Drip builds a reliable transport layer on top of the Trickle algorithm.

4.5 Time Synchronization

It is necessary to apply a time synchronization protocol in order to make sure that all nodes in the network have a common understanding of the global time. Flooding Time Synchronization Protocol (FTSP) [17] is an example of a protocol with this capability that is designed for use in WSNs. For our purpose, this capability is particularly important for the road surveillance scenario. Since the velocity of the passing vehicles can be potentially very high it is important to determine the precise timestamp of the detection.

4.6 Localization

Measurement data from the battlefield is meaningless without a precise knowledge of the location from where the data was captured. The intuitive solution to sensor network localization would be to equip every sensing node with a GPS. However, we have designed and implemented a simpler
Table 4.1  Software components in SASS. Memory requirements are given in Bytes. TelosB has 48K Flash and 10kB RAM in total

<table>
<thead>
<tr>
<th></th>
<th>Component</th>
<th>Flash</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocols</td>
<td>CTP</td>
<td>9674</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Drip</td>
<td>3780</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td>Ftsp</td>
<td>7524</td>
<td>228</td>
</tr>
<tr>
<td></td>
<td>Deluge</td>
<td>30190</td>
<td>975</td>
</tr>
<tr>
<td></td>
<td>LPL</td>
<td>1702</td>
<td>58</td>
</tr>
<tr>
<td>Sensors</td>
<td>Ir</td>
<td>3488</td>
<td>720</td>
</tr>
<tr>
<td></td>
<td>Radar</td>
<td>4618</td>
<td>714</td>
</tr>
<tr>
<td></td>
<td>Acoustic</td>
<td>2754</td>
<td>1158</td>
</tr>
<tr>
<td></td>
<td>TinyOS</td>
<td>16558</td>
<td>981</td>
</tr>
</tbody>
</table>

solution to save hardware cost and reduce the system complexity. In our scenarios, the sensors are deployed manually in the field by a walking soldier. We can take advantage of this fact and implement a mobile localization solution (cf. Fig. 4.1). The mobile node is fitted in the rucksack of the soldier deploying the network and consists of a computer with a high precision GPS connected to a TelosB mote. The mobile node transmits its location on 5s intervals. We ensure that only one node receives the broadcast beacon packet by transmitting the beacon with -25dBm output power, limiting the range to about 3m. The sensing node considers only the first beacon it receives after startup. Beacons received after this stage are silently ignored. The location error with this approach would be similar to the location error given by the GPS +/- 1m (depending on the mobile node trajectory).

4.7 Software Considerations

TelosB is limited to 48kB of flash and 10kB of RAM. Although platforms with larger capacities exist, both the cost and the energy consumption raises quickly with more advanced microcontrollers. Since memory is a limiting resource, it is necessary to prioritize between the desired software components. Table 4.1 shows the memory requirements in flash and RAM for the different software components.

Initially, it was a strong desire to incorporate Deluge to the SASSNode. Deluge [18] is a reliable data dissemination protocol for propagating software images. It is very attractive in our design since it allow reprogramming of the entire WSN after the nodes are deployed in the field. Due to the footprints of the mandatory components such as the operating system, CTP, Drip, FTSP and the sensor algorithms, auxiliary and non-critical components such as Deluge and LPL had to be left out.
5 Field Test Experiments - Perimeter Surveillance

5.1 Experimental setup

The surveillance experiments were conducted at the Norwegian Army Combat Maneuver Training Centre (NACMTC) at Rena, Norway. A small Forward Operating Base (FOB) was established at a forest edge. The salient threats to such a base include enemy reconnaissance attempts or IED placements. Such attacks would most likely be performed via the forest. Forest areas with ditches and dense vegetation, are not easily monitored by an elevated sensor platform inside the FOB and is ideal for Wireless Sensor Network instrumentation. Hence, this area was instrumented with our 50-node sensor network. With respect to Fig. 1.1, we only instrumented one side of the FOB. In addition, we instrumented this side with a state of the art UGS, UMRAmini, alongside with our sensor system for comparison. UMRAmini is manufactured by Exensor [19] and the kit consists of six seismic detectors connected to a base station via a wireless mesh network. In addition to the two sensor networks, the FOB was protected by a team of soldiers in an observation post. The soldiers were equipped with night vision capabilities.

Four highly-trained soldier teams played the roles as insurgents instructed to conduct reconnaissance of the FOB. The insurgents were equipped with NACMTC combat training equipment allowing their movements to be evaluated after the exercise by incorporating GPS position tracking. This enabled us to precisely investigate the number of positive and missing detections and false alarms after trial completion. The multiday test was split in four trials, one trial for each of the insurgent teams. Two of the trials (D1/D2) was conducted during day-time and two (N1/N2) during the night.

\[\text{Figure 5.1} \quad \text{The real tracks of the intruders walking towards the FOB (red traces going from right to left) compared with the sensors producing alarms}\]

The SASSnodes were configured to send an alarm when the activity measured on one of the sensors was above a certain threshold value. To filter out erroneous measurements the central software was configured such that alarms were only transmitted to the user when both PIR and radar had been active on the same SASSnode within a time interval of 5s.
5.2 Results

It became clear during the daytime experiments (D1/D2) that the sensor algorithms were incapable of filtering out background noise caused by wind blown grass and trees. During these two experiments the average wind speed recorded was 3.3 and 8.1 m/s respectively with wind gusts estimated to be up to the double. During the wind gusts, all vegetation and trees were moving in the detection sector of the sensors. Both IR and radar were affected by this problem and a vast number of false alarms were collected by the central software. Although we were able to remove some of the erroneous alarms by using spatio-temporal filtering and fine-tuning of the key parameters for the IR and radar algorithms, the results were inconclusive.

For the nighttime experiments (N1/N2) the sensor network gave very promising results. During these two experiments, the wind speed was below 2 m/s. This reduced the background-noise and improved the quality of detection dramatically. Fig. 5.1 shows the alarm reports from the sensors (with filtering) during these two experiments related to the real movement of the intruders. It was possible to track the movements of intruders through the network with high accuracy. The false alarm rate was in this case zero when we employed spatio-temporal filtering. As an example of this, the N2-experiment gave only one false alarm from a single sensor node during the experiment. This alarm was not correlated with nearby sensors and was easily filtered out in the software. Fig. 5.1 also demonstrates the negative effect of using a strict filtering, since there are several sensor nodes here that did not produce alarms even if the intruders were very close to the sensor. The negative detections are caused partially by the limited sensing range and narrow detection sector of the sensors themselves and partially by the placement of the nodes, which were camouflaged and hidden.

<table>
<thead>
<tr>
<th>Surveillance-system</th>
<th>Trial N1</th>
<th>Trial N2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(t_d)</td>
<td>(N_s)</td>
</tr>
<tr>
<td>Observation soldiers</td>
<td>(\infty)</td>
<td>0</td>
</tr>
<tr>
<td>UMRAmini</td>
<td>(\infty)</td>
<td>0</td>
</tr>
<tr>
<td>SASS</td>
<td>17</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.1 shows the detection time and the number of sensors involved in the detection for the nighttime trials. The table shows the results for the observation post soldiers, the UMRAmini system, and for our SASS system. The detection time is defined as the time difference between when the time the invaders came into the area and the time of detection. The SASS system had the best performance regarding both the detection time and the number of sensors involved. Admittedly, the high number of SASS sensors is the main reason for this result. Although it is highly probable that similar results would have been obtained for the UMRAmini system using an equivalent number of sensors this is a more costly alternative. Due to the limited visibility in the area, it is unlikely that an increased number of soldiers in observation posts would have improved the result.
6 Field test experiments - Road Surveillance

6.1 Experimental setup

In this section, we describe road-surveillance using our sensor network (cf. Fig. 1.1). 40 SASSnodes were placed along the roadside with an inter-distance of 10m. 20 nodes were placed on each side of the road, covering a total road length of 190m. The limited available memory on the TelosB node made it difficult to use all three sensors simultaneously. Hence, the 20 nodes on one side of the road were programmed to use acoustic sensors, while the remaining 20 nodes on the opposite side of the road were programmed with radar and PIR. Fig. 6.1 shows the road and the placement of the sensors. We performed a series of experiments with the passage of vehicles with and without stop.

![Monitoring of a 190m road with 40 sensing nodes equipped with IR, radar and acoustic sensors.](image)

6.2 Results

For each SASSNode, alarms can be produced either based on one single sensor reading or based on the combination of simultaneous readings from both PIR and radar. It is also possible to combine sensor readings from a SASSNode with PIR and radar on one side of the road with the microphone-enabled SASSNode on the opposite side of the road. In the experiments, both the PIR and the microphone produced some false alarms, mainly due to the weather conditions (wind, which created noise, and changing clouds, which is a challenging condition for the PIR). The radar detector had excellent performance in detecting the vehicles.

Fig. 6.2 show an example of one of the trials. Here, a vehicle enters the monitored area, then stops in 60s, before it continues. In this figure, only the alarms from the radar detector are used. The system can estimate both the speed of the vehicle; determine the exact position where it stops, and the duration of the stop. Hence, it is possible to use this information to separate suspicious behavior from normal behavior. I.e., a vehicle that stops immediately leads to an alarm but vehicles that drive through the area are accepted.
Figure 6.2 Using radars to detect the passing of vehicles along a road. The figure compares the time and speed calculated by FTSP-time with real GPS traces from the vehicle.

7 Network performance

In this section we evaluate the performance of the CTP protocol in SASS. Three key performance metrics are evaluated: Packet delivery ratio (PDR), churn and the number of hops. PDR is a primary metric for network performance and gives insight on the reliability of the system. A PDR of at least 95% is desirable. Churn is the average number of parent changes per hour per node, and is a measure of the network dynamics. A high churn implies that the links are not stable, meaning that the control traffic necessary to maintain the routes is increased. The average and maximum number of hops in the topology gives a picture on the network density and node distribution.

To provide a variation in network topology and RF environment, the performance evaluation includes six different network topologies. These are presented in Table 7.1. The first experiments were performed indoor to prepare for the outdoor real deployments. There are several reports on sensor networks that perform well in controlled situations yet poorly in practice. Hence, we attempted to create our indoor testbed with a high degree of realism. For this purpose we designed a 20-node wall-mounted network that allowed induced RF-interference in a controllable way. Experiments were conducted both with and without interference. The setup with interference let us test for the worst case before deploying a large network in the field. The third topology involved 50 SASSnodes over a three storage office building. The fourth configuration was the perimeter surveillance deployment described previously. Finally we report on results from the road surveillance experiments with two different sink locations.

The use of different topologies gives an important insight in the similarities and differences between small-scale testing and real-life deployments. Table 7.1 shows that the non-interfered wall mounted...
### Table 7.1  The network performance. Churn is the average number of parent changes per hour and PDR is the average packet delivery rate

<table>
<thead>
<tr>
<th>Network deployment</th>
<th>Size</th>
<th>Nodes</th>
<th>Hops</th>
<th>Churn</th>
<th>PDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m^2$ or $m^3$</td>
<td></td>
<td>avg</td>
<td>max</td>
<td></td>
</tr>
<tr>
<td>Wall -25dBm</td>
<td>2.5x2.5</td>
<td>20</td>
<td>2.0</td>
<td>4.1</td>
<td>0.46</td>
</tr>
<tr>
<td>Wall -25dBm (w. interference)</td>
<td>2.5x2.5</td>
<td>20</td>
<td>6.9</td>
<td>180</td>
<td>17.7</td>
</tr>
<tr>
<td>Office area</td>
<td>40x70x6</td>
<td>50</td>
<td>3.0</td>
<td>6.0</td>
<td>0.72</td>
</tr>
<tr>
<td>Perimeter surveillance</td>
<td>80x120</td>
<td>50</td>
<td>3.0</td>
<td>5.1</td>
<td>0.96</td>
</tr>
<tr>
<td>Road surveillance (sink at the end)</td>
<td>190x4</td>
<td>40</td>
<td>6.5</td>
<td>11.2</td>
<td>3.64</td>
</tr>
<tr>
<td>Road surveillance (sink at the middle)</td>
<td>190x4</td>
<td>40</td>
<td>3.7</td>
<td>9.0</td>
<td>1.38</td>
</tr>
</tbody>
</table>

1Loops occurred frequently, causing the maximum number of hops to be very high.

network slightly overestimated the PDR and underestimated churn compared to the real deployments. With interference added, the PDR decreased dramatically, and was comparable to the road surveillance deployment with the sink at the end. The indoor office deployment was topologically very similar to the perimeter surveillance deployment, and the results for both PDR and churn matched quite well, albeit they were slightly optimistically for the indoor case. The road surveillance results show that the placement of the sink is crucial. In Fig. 7.1 we study the packet delivery in detail for both the end-placed sink and middle-placed sink topology. Using an end-placed sink, there is a correlation between the number of hops to the sink, and the packet loss rate. All nodes are, however, affected by poor network performance. The high churn in the network contributes to an overall reduced performance due to topology inconsistencies and a vast number of beacon messages.

![Figure 7.1 The distance to the sink (in hops) vs. the Packet Delivery Ratio (PDR)](image)

(a) Sink located at the end

(b) Sink located at the middle

The results from the road surveillance deployment show that the network performance suffers when the nodes are sparsely spread along a road. By placing the sink in the middle of the network, the packet delivery is improved. In some scenarios, however, the optimal placement of the sink might be difficult to determine. Another and more flexible method to improve the performance is to use radios
with longer range. This will reduce the average number of hops and increase the network density.

A final conclusion from the experiments is that it is difficult to create an indoor testbed that fully captures the outdoor target scenario. One reason for this is that there are dynamics in the real-world that does not exist in a testbed. However, our results reveal that wisely induced interference can give an indication on how the protocols deals with dynamics in the field even if the induced dynamics are exaggerated.

8 Lessons Learned

Conducting real-life experimentations and evaluation of large sensor networks involves significant cost in equipment and man-hours. Our experiences in developing and testing SASS revealed a series of lessons learned which are valuable for planning our future test campaigns.

Establish the ground truth: An important part of evaluating a WSN is to compare its output with the ground truth. In our case, this was done by comparing the alarms produced by the WSN with GPS tracks of personnel and vehicles in the monitored area. In addition, we used a separate UGS as a logger system alongside with our WSN. However, animals and moving trees and bushes out of our control created false alarms making it difficult to quantify the exact performance of the WSN. We also overlooked the effect of weather conditions on the sensor performance. Although a stationary weather station was located within 500m of the WSN providing approximate weather data, we should have brought our own weather station to provide detailed information about wind speed, sun conditions, temperatures and humidity.

Don’t forget the “S” in the WSN: The vision of large networks with small disposable nodes means that the sensors must be of low cost, with obvious compromises on the quality. The poor performance on the individual sensors can to a certain extent be compensated by using a large amount of sensors and by combining two or more detector types (i.e., IR and radar). Admittedly, we prioritized the task of ensuring reliable sensing below that of creating a functional wireless network. This decision had obvious consequences and resulted in a vast number of false alarms when the weather conditions were disadvantageous. The sensing task would have been more successful with higher quality on the sensors and better algorithms.

Choose the right hardware platform: Most of the successful WSNs in the literature use the TelosB or similar platforms. Hence, we took it for granted that this unit would be suitable for our needs as well. There were, however, two limitations on the TelosB that we did not consider. First, during the design phase it did not occur to us that we would reach the code limit of the TelosB platform. All the features we would like to add, quickly added up to more than the size of the TelosB flash. Second, the radio range on the CC2420 transceiver is very limited. This led to some challenges in designing realistic experiments. A radio with longer range that the TelosB provides can give a more flexible and useful WSN. It is also much simpler to design a routing protocol that ensures reliable communication over 2-5 hops than over 20 hops. To conclude, a radio with longer range would have simplified the protocols as well as given higher flexibility in conducting experimentations.
Obtain node status: Debugging a WSN is intrinsically hard due to the distributed and autonomous design. Our nodes had the ability to transmit regular status messages. This gave us a basic overview of the status on the network and the sensor settings for each node. However, in some circumstances it would have been desirable to fetch raw sensor data for debugging purposes. In addition, there were some episodes when we lost connection with one or more nodes for a period of time. Since the network was down, no status messages were received by the sink. Further, since no logging was performed locally on the nodes, the precise cause of these error conditions were impossible to identify in the field.

9 Conclusion

Military base protection is a challenging type of surveillance that is ideal for Wireless Sensor Networks (WSNs). We have developed a Situational Awareness Sensor System (SASS), which consisted of 50 TelosB wireless sensor nodes with radar detectors and passive IR sensors. We ran several experiments with our sensor network deployed in a military training facility to quantify the operative effect as well as measuring the capacities and challenges of the networking system. The system was compared to a state-of-the-art surveillance UGS as well as soldier observation posts at the base. The SASS system provided high efficiency in detecting adversary personnel and vehicles in perimeter and road surveillance scenarios.

The major limitation with the system was the poor performance on the individual sensors used on the nodes. The system presented in this report is still under development, and we recognize some unresolved issues. Future work could therefore include a) weather proofing and miniaturization of the sensor enclosure, b) a power management scheme, c) increasing the radio range, d) investigating better sensors and algorithms.

The report identify the key lessons learned, which were: the importance of knowing the ground truth; that it is paramount to use high performing sensors and a WSN platform that suits the target application; and the significance in logging all information produced on each sensor node.

Acknowledgment

The work in this report was financed by the Norwegian Defence Research Establishment and the University Graduate Centre Kjeller (UNIK). We owe great thanks to Joakim Myren for help with engineering the sensing nodes. The authors would also like to acknowledge the support from the Norwegian Army Combat Maneuver Training Centre for assisting the field test experiments.
References


