WASP

Lessons learned from application prototype experiments in WASP

Deliverable D6.7

Public Report

Project Number	:	IST-034963
Project Title	:	WASP
Deliverable Type	:	Report

Deliverable Number	:	D6.7
Title of Deliverable	:	Lessons learned form application prototype experiments in WASP
Nature of Deliverable	:	Public Report
Internal Document Number	:	WASP_D6.7_lessons_application_prototypes.doc
Contractual Delivery Date	:	August 2010
Actual Delivery Date	:	September 2010
Contributing WPs	:	WP 6
Author(s)	:	Kees Lokhorst, Louis Attalah, Alessio Corongiu, Pieter Hogewerf, Rudi de Mol, Angelo Colabufo, Alberto Bosisio, Laurent Gomez, Mario Schuster, Lasse Thiem, Francois Ingelrest, Richard Verhoeven, Martijn Bennebroek, Andreas Lachenman, Marcel Steine, Jacopo Sarri, Bert Ipema

Abstract

This deliverable describes the final application prototypes built and experiments and simulations performed to validate the generic WASP system approach. Real sensor deployments have been made for relevant Elderly Care and Herd Control application scenarios in the test bed facilities in London and Lelystad, respectively. For Automotive, simulations have been used for in-vehicle and tunnel management scenarios.

For Elderly Care, the challenges and the lessons learned are discussed around the topics of: ease of use of system, reliability issues and difficulties of debugging, wearability of nodes, flexibility in programming models and usability in patient trials. The lessons learned in the Herd Control environment focus on programming and dynamic service uploads of nodes, state-based triggering and on-node classification of sensor data, communication tools and location awareness. The development of integrated demonstrators and, especially, of the Elderly Care and Herd Control test beds proved extremely useful to identify various technical and practical issues that typically emerge only during actual deployments. Test bed prototyping proved a good driver to guide technical alignment and successful integration within a large IP project like WASP whereas the application and generic system requirements played an important role as well.

For automotive,simulations for in-vehicle scenarios reveal that the (star-based) WASP network solution developed for Elderly Care provides very good QoS performances in terms of high packet delivery ratios (close to 99%) and low latencies (in the order of tens of milliseconds) but less attractive in terms of power consumption and, hence, autonomy of battery-powered nodes. Similar simulation results have been obtained for tunnel management scenarios where use is made of the (mesh-based) WASP network solution developed for Herd Control. When replacing the (always-listening) IEEE802.15.4MAC by (duty-cycled) WiseMAC in the network stack, significant improvements in energy-efficiency can be achieved although at a degradation in latency and delivery ratio.

The current integrated WASP solution demonstrated in the herd control and elderly care test beds can be well used as a research tool for animal and human behaviour scientists at ICL and WUR to enable data acquisition for period of several weeks to study behaviour and, in the end, come up with better models and treatments. This understanding is essential to quantify benefits and costs and, thereby, to motivate any next steps from the current WASP research technology prototypes towards industrial application prototypes and, subsequent, commercial solutions.

Keyword list

Prototyping, Elderly Care, Herd Control, Road Management, Simulation, Lessons Learned

Table of Contents

Table of Contents 4

1 Introduction 6

2 Elderly Care Prototype 8

2.1 Introduction 8

2.2 Case description of EC scenarios 9

2.2.1 Ease of deployment 10
2.2.2 Monitoring of Activities of daily living (ADL) 11
2.2.3 Hypothermia detection 14
2.2.4 Cardiac monitoring 14
2.2.5 Fall detection 14
2.2.6 Scalability 15

2.3 Algorithm and software development 15

2.3.1 On-node Classifier 15
2.3.2 Alarming 17
2.3.3 The use of ambient and wearable sensors 17
2.3.4 On-node ECG monitoring 24
2.3.5 User-interfaces 25

2.4 Discussion and lessons learned 31

3 Herd Control Prototype 33

3.1 Prototype description 33

3.2 Prototype development and testing 39

3.2.1 Hardware experience 39
3.2.2 Ease of deployment (energy efficiency) 39
3.2.3 Network stability and scalability 40
3.2.4 Gateway integration 42
3.2.5 Herd Control algorithm and location awareness 44
3.2.6 EIC User interface 47
Step duration per node 48
3.2.7 Prototyping and testing 53

3.3 Herd Control discussion 53

4 Simulations of automotive scenarios 57

4.1 Scenarios description 57

4.1.1 Wireless sensors and data replication within a tunnel 57
4.1.2 In-vehicle sensor networks 60

4.2 Models and tools 62

4.2.1 Evaluation metrics 62

4.2.2 Tools 63
4.2.3 Models 63

4.3 Simulation parameters 65

4.3.1 Settings for Road Tunnel Scenario 65
4.3.2 Settings for In-vehicle sensor networks scenarios 72

4.4 Results 78

4.4.1 Results analysis 79
4.4.2 In-vehicle scenarios 80
4.4.3 Road tunnel scenario 82

4.5 Benchmarking 85

4.5.1 Motivation 85
4.5.2 Application for tyre pressure monitoring system 86
4.5.3 In-car Wireless Sensor Networks 88
4.5.4 Power consumption of currently available platforms 89
4.5.5 Cost/benefit budget 89

4.6 Road Management lessons learned 89

5 Requirement fulfilment 91

6 Concluding remarks 93

References 95

1 Introduction

The WASP project works on the development of a new generation of Wireless Sensor Networks by covering the whole range from basic hardware, sensors, processor, communication, through the packaging of the nodes, the organisation of the nodes, towards the information distribution and a selection of applications. The general goal of the research project WASP (acronym of Wirelessly Accessible Sensor Populations) is the provision of a complete system view for building large populations of collaborating objects.

WASP includes research on applications because the properties of the required service will influence the configuration of both sensor network and application for optimum efficiency and functionality. The applications are sought in three business areas (Figure 1.1):

- Assisted Road management lowers the risk on accidents, reduces pollutions, reduces petrol consumption, and increases the efficiency of our roads (e.g. in vehicle and road side applications).
- Elderly Care benefits by reducing medical costs and increasing the autonomy of the elderly (behaviour monitoring, e.g. fall detection and risk prevention; health monitoring, e.g. heart failure, COPD).
- Herd Control reduces the health-risks, which threaten the cattle industry in the European area (e.g. activity and lying behaviour).

These business areas have been selected for their societal relevance and their expected technical differences. The expectation is that when the WASP concept will work in these three business areas it will also work in other business areas too.

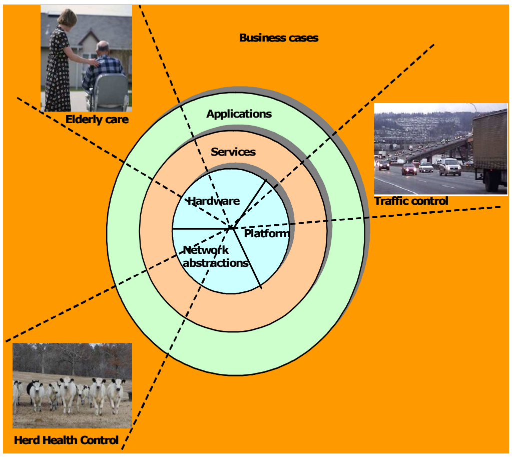


Figure 1.1 - Business areas in the WASP project

The WASP project is primarily oriented on the technological development of a new concept for collaborating sensor networks. The mission of the WASP project is that 1) the WASP project provides theory, methods, hardware and software to construct highly optimised applications on a network of generic and flexible nodes and 2) the WASP project enhances industrial acceptance of the technology for collaborating objects, by deploying the technology in a steadily increasing number of business areas. Translated in a more popular way the 'promises' of the WASP concept are:

- Smart wireless multi sensor network (WSN) with wireless access to 'Internet environment'.
- State based triggering of sensors and WSN.
- Smart energy, security, accuracy, storage and processing strategies within WSN and through services available in the neighbourhood.
- Moving objects becomes central in real time environment (time and location).
- User friendly application development environment.

To reach these goals the WASP project is divided into the work packages, see also Figure 1.2). In this report the overall results of WP6 are reported. The objective of the 'Applications' work package is to provide real problem areas for the embedded system area of wireless accessible sensor networks. Two scenarios, selected for their societal relevance and technical differences, representing general and specific requirements for the sensor networks, are used to motivate and guide the design of the components, to guide testing of components and perform tests on system level by using two test bed situations, to develop prototypes and to set up security patterns. The second objective is to support WP7 in involving SME's and end users by showing them the possibilities of future applications, which can be based on wireless sensor networks that operate autonomously in an intelligent infrastructure (realizing two prototypes). Part of this objective is the test and validation of the sensor network design and the genericity of the offered design in the three business areas.

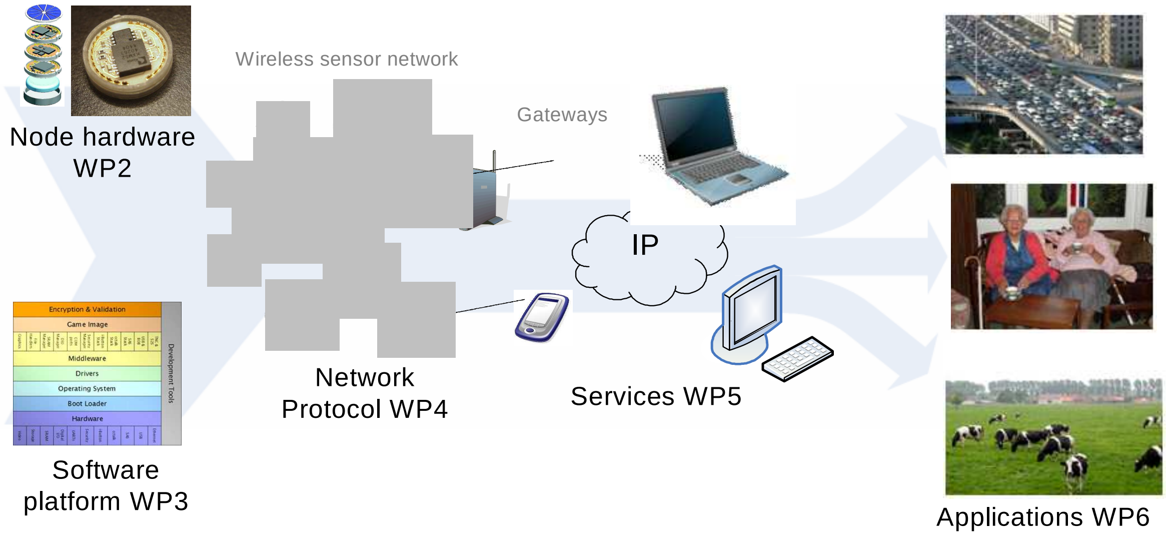


Figure 1.2 - Work packages in the WASP project

The prototype selection resulted in the creation of test bed situations and prototypes for the elderly care business area with the scenario EC1: '*Activities of daily living using wearable/ambient sensors'*, and the herd control business area with the scenario HC2: *'Detection of health problems with focus on claw health and locomotion'*. These two scenarios are used for prototyping and testing at system level. The results will be reported in Chapter 2 and 3 respectively. Some of the Road management scenarios are used in a simulation environment to test the reusability and genericity of the WASP concept. This is reported in Chapter 4. In Chapter 5 and 6 the overall conclusions with respect to the generic requirements and the experience with the WASP concept are discussed.

2 Elderly Care Prototype

2.1 Introduction

The choice of elderly care as a WASP test-bed application is motivated by the steady decrease of the health-workers to retiree's ratio and the miniaturization of technology allowing sensors to be pervasively woven into the patients' surroundings without affecting their daily routines. This domain is of growing societal relevance as the proportion of people aged 60-plus is projected to increase rapidly in the next two decades, combined with an increase in the number of patients having chronic diseases, including cardiovascular diseases, diabetes and chronic respiratory disorders (such as COPD, Chronic Obstructive Pulmonary Disorder). The use of wireless sensors for elderly care could enable remote monitoring that allows care providers and clinicians to observe patients' health continuously instead of obtaining snap-shot assessments as currently being done during hospital or general practitioner visits. It could also provide alerts in case of symptom deterioration or a perceived threat, such as that of

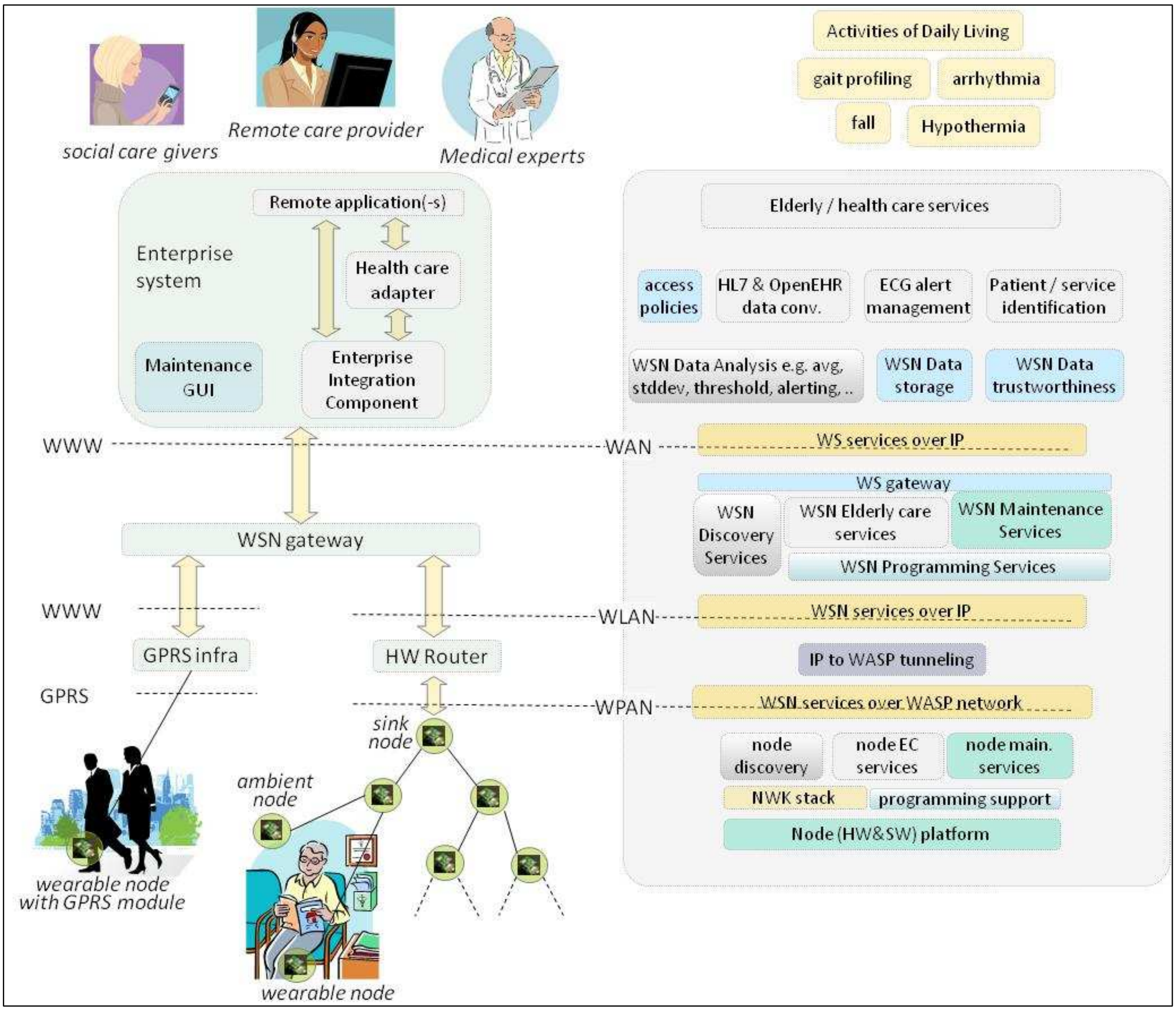


Figure 2.1 - Overview of the WASP architecture tuned for elderly care monitoring. The left hand side shows the hierarchical layers and physical deployment whereas the functionality associated with each layer is shown on the right hand side. The GPRS connection is shown for completeness but will likely be not integrated in the elderly care test bed. The GPRS module is available but has been demonstrated in the herd health control test bed of WASP.

Hypothermia [2.1] or long periods of inactivity. WASP addresses remote elderly monitoring with a special focus on activities of daily living that are important in assessing the decline of cognitive and physical functions over time and on cardiac ECG monitoring. Figure 2.1 depicts the overall WASP architecture that will be described in this section along its main hierarchical layers and associated functionalities illustrated on the left and right hand sides, respectively. The architecture itself is generic and has been defined in the first year of WASP by studying application requirements for three domains (elderly care, heard health, and automotive) that have been selected for their societal relevance and potential benefit from WSN-based solutions. The common requirements include reliability, sensor node life-time (energy-efficiency), ease of deployment, end-to-end communication with the back-end and synchronization of data from various sensor modalities. More specific elderly care requirements include the support of mobility, the ability to remotely set thresholds and alarms, privacy and security of patient-sensitive data and the ability to generate patient-specific reports to observe patient behaviour over time.

The bottom (left) part of Figure 2 illustrates that the lowest hierarchical layer contains wearable and ambient wireless sensor nodes. Here, wearable nodes allow to continuously monitor relevant parameters of an elderly person in indoor and outdoor settings whereas ambient nodes are typically used to monitor environmental parameters within different locations in buildings and, moreover, to relay wireless messages between sensor nodes and sink nodes. Indoor sink nodes interface with the next (intermediate) hierarchical layer containing a HW router, that bridges the WPAN communication between nodes to (W)LAN communication, and a WSN gateway, that interacts with wireless sensor nodes through a set of WSN service messages. For outdoor communication, a GPRS module can be activated to provide an alternative connection to the WSN gateway. Finally, the highest hierarchical layer contains the enterprise (or backend) system that interacts through a set of (XML-based) web-services with the WSN gateway. The enterprise system provides end-users, depending on their data access rights, useful means to analyze the collected WSN data and to receive alert signals for specific conditions.

As mentioned, the right-hand side of Figure 2.2 depicts the distribution of functionalities across the hierarchical layers. On the highest layer, the so-called Enterprise Integration Component acts as a generic mediation layer between remote (enterprise) applications and WSNs. This is achieved by providing functions such as storing WSN data, assessing quality (trustworthiness), analyzing trends, and generating events (including alerts) for specified conditions. The system also includes a Healthcare Adaptor that acts as a healthcare-specific mediation layer converting WSN data to OpenEHR format enabling HL7-compliant communication with Healthcare systems.

2.2 Case description of EC scenarios

Deliverable 6.3 [2.2] highlighted the path taken for an integrated roadmap within the WASP project and within the EC test bed in particular. Partners agreed on several relevant scenarios that are briefly summarised in the next section and then followed a road-map for integrating components leading to these scenarios.

The scenarios highlight several important issues that are relevant for elderly monitoring including energy efficiency, integration of heterogeneous sensors, mobility, privacy and the possibility of generating alerts.

Deliverable 6.3 also introduced the ‘physical’ test bed at ICL where experiments will be taking place as well as the sensors used and the parameters of importance. These parameters were selected after confirming with medical staff working closely with the group at ICL about the relevance of the demos presented.

The next section briefly describes the scenarios selected in order to give the reader an introduction and a better understanding of the technical challenges described in the next few sections.

2.2.1 Ease of deployment

This scenario consists of a preliminary step in order to move on to more elaborate scenarios and make sure that the nodes are reliable and can be used for the suggested demos. The following parameters are to be shown:

1. **Node monitor**: Show service installation upon node power up. This would allow users to detect if all services are installed on nodes. If a node switches off or reboots, this would also be shown in the monitor as the node (and its services) would quit the list, then join once all services are back.
2. **Topology monitor**: It is important to have a tool that can show network topology which as patients will be moving and we would like to observe how well the system copes with the change of distances and network structures.
3. **Gateway Visualiser**: The gateway also has a visualisation screen that enables the user to observe the signals coming in from the nodes in real-time.
4. **Maintenance GUI services**: These services are useful for person tracking, providing a connectivity graph and showing battery status per node.
5. **Statistics service**: Enabling the user to observe packet delivery ratio as well as other statistical measures that would be useful to monitor the performance of the system.

Figure 2.2 shows the structure of this scenario and some of the components that will be integrated within this scenario. Figure 2.3 shows the node monitor in more detail as well as the gateway visualisation screen.

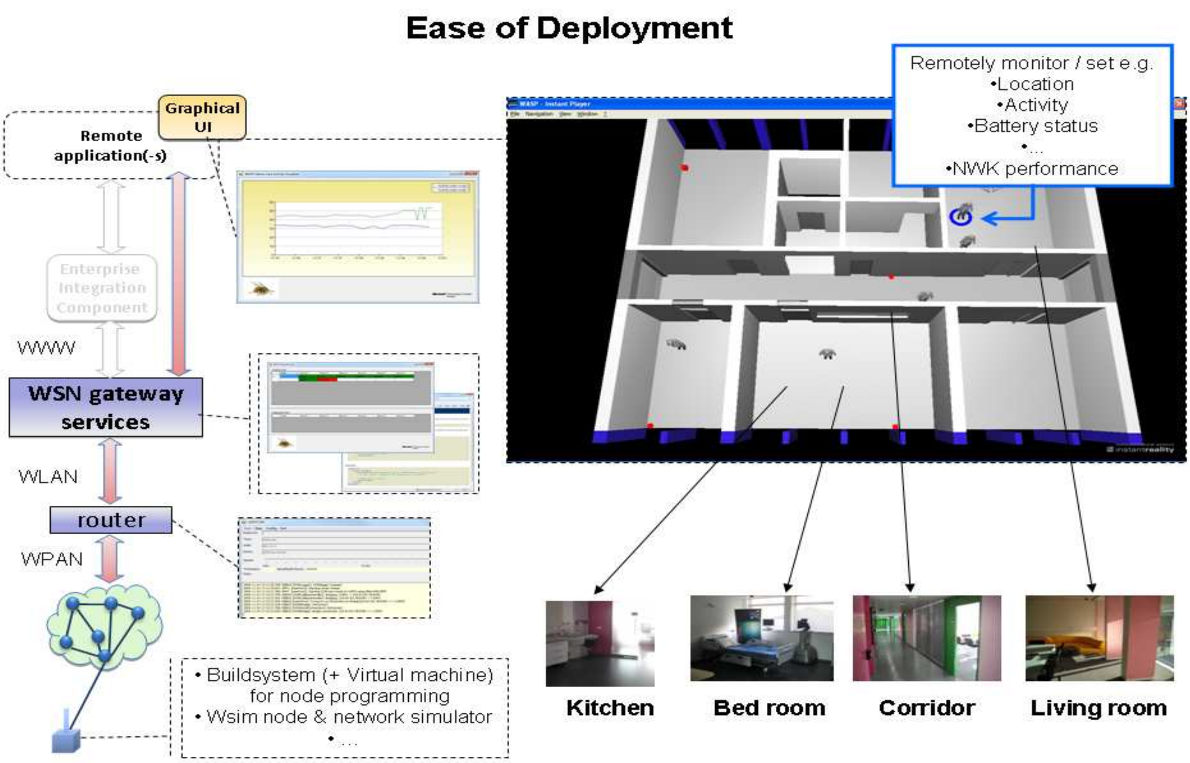
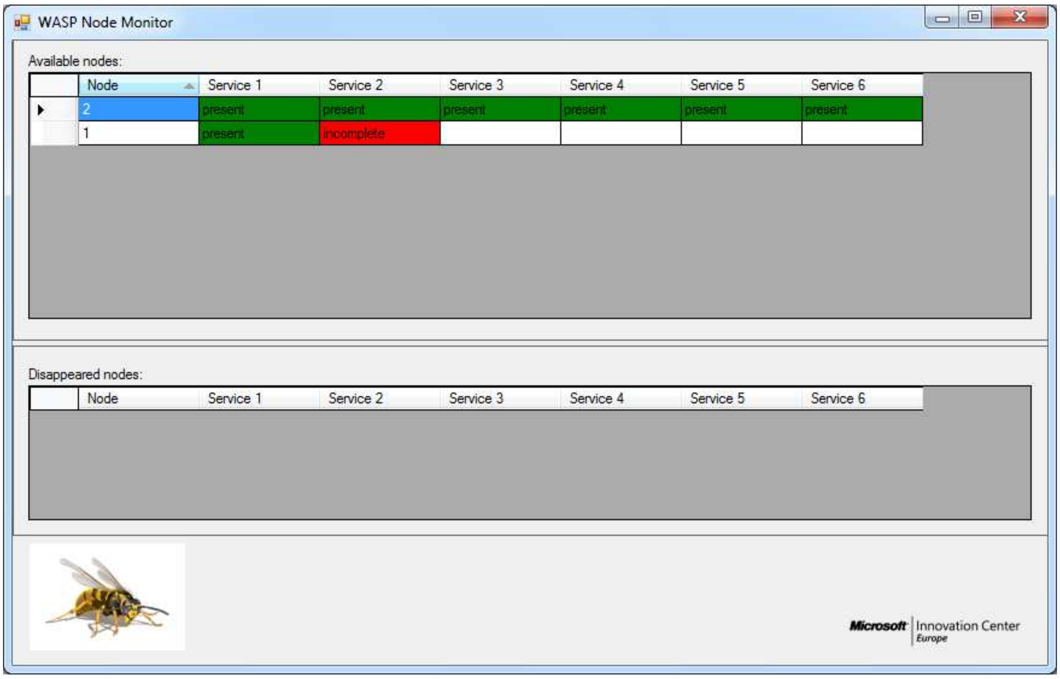
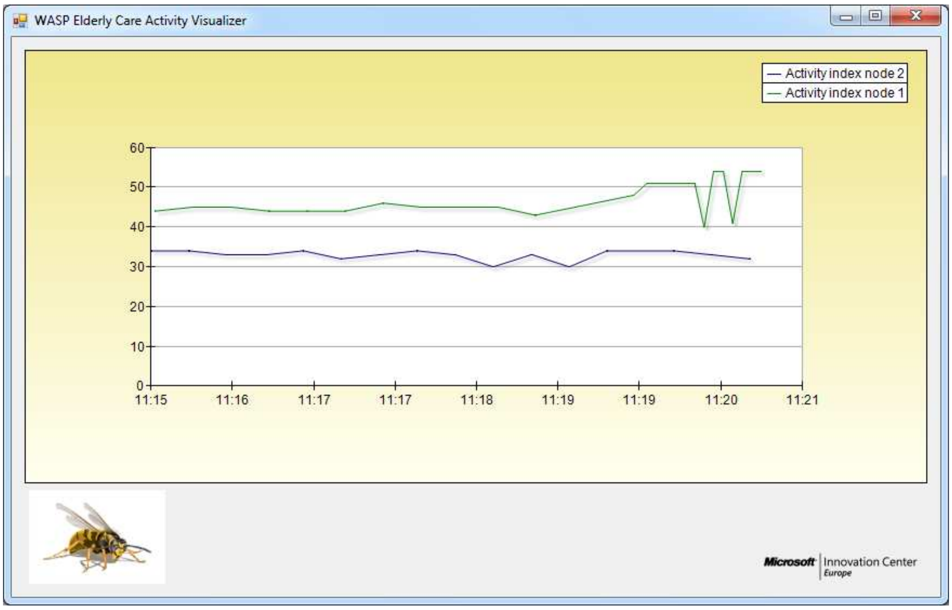


Figure 2.2 - Showing a plan of the test bed as given in the maintenance GUI and the structure of the system to be tested in the ease of deployment scenario.



(a)



(b)

Figure 2.3 - (a) The node monitor showing nodes connecting and updating the services available on each node and (b) showing the gateway visualiser.

2.2.2 Monitoring of Activities of daily living (ADL)

Individuals who are active are almost two times less likely to die prematurely from a heart attack than their sedentary counterparts [2.3]. There is strong evidence to suggest that people who are physically active can reduce the risk of developing coronary heart disease, stroke and type II diabetes by up to 50% and alleviate the risk of premature death by about 20-30% [2.4].

Objective measurement of physical activity in free living conditions without behaviour modification plays an important role in studies providing intervention strategies to increase daily activity [2.4].

Moreover, understanding the relationship between activity levels, disease progression and medication requires person-specific behaviour profiling that can be attained by observing activity changes over time and relating them to health and well-being. Deterioration in conditions such as heart arrhythmias [2.5], diabetes mellitus and hypertension (high blood pressure affecting approximately 50 million individuals in the United States alone [2.6]) is gradual, and often subtle over time. For these conditions, continuous observation of activity levels as well as physiological parameters can provide more accurate diagnosis and tailored treatments for each subject, compared to the currently used infrequent, snap-shot diagnosis and management. Given the importance of the topic of activity monitoring for healthcare providers over the world, the WASP project aims to focus on some scenarios that are of relevance to this area.

2.2.2.1 Activity detection using wearable nodes

This scenario aims to use wearable nodes within a home environment to look at activities of daily living of several subjects. Issues such as mobility will be catered for and an activity classifier will be developed on the node to minimise the amount of data that is sent to the backend. The activity classifier is described in detail further on in this document.

An activity index is also developed which can be described as the average variance over 3 accelerometer axes over a given period of time (we normally use a time window of up to 5 sec). This index is shown in Figure 2.4 for a range of activities.

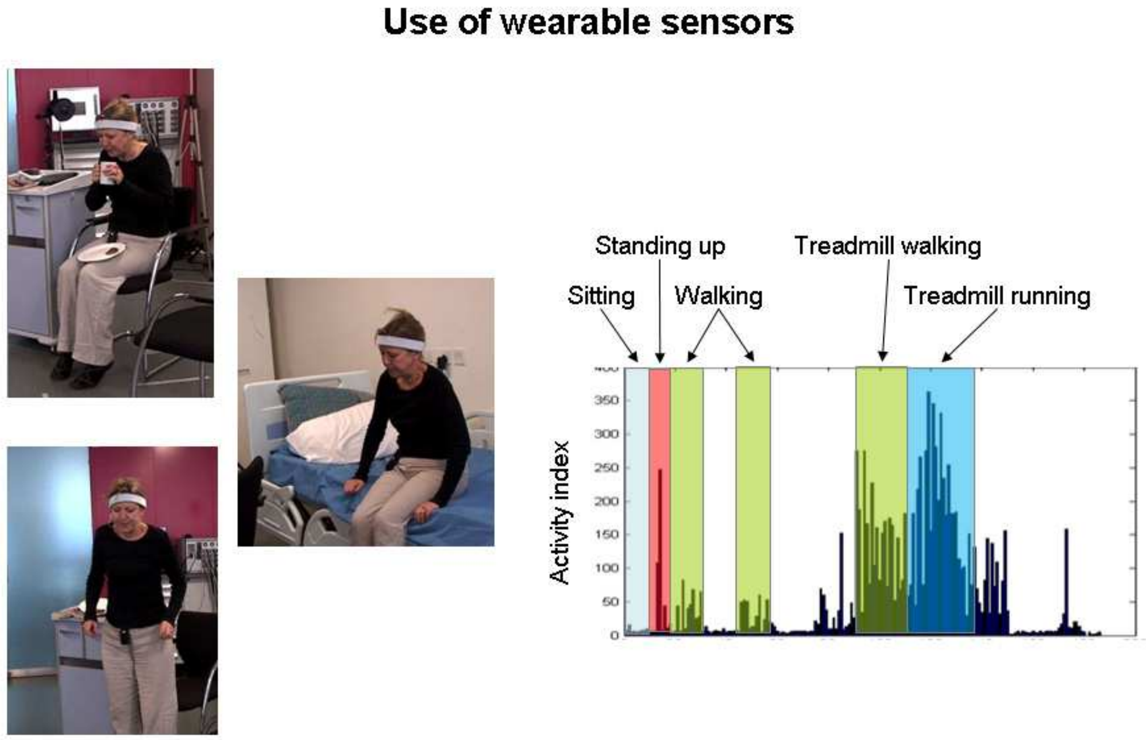


Figure 2.4 - The use of wearable sensors to observe activities of daily living. The activity index is shown on the right with different values for different activities.

2.2.2.2 Use of ambient and wearable sensors

This scenario proposes to combine information obtained from wearable and ambient sensors. Data from both modalities could provide more context-awareness as activity levels could be correlate with location and information about the surrounding. For this work, we will be using a combination of wearable WASP nodes with the Housevibe radar. Housevibe is a radar-based monitoring system that provides information on human movement in a given indoor area. Radar is an attractive technology for long term monitoring of humans because it does not need to be carried by the user, can be placed behind walls and is able to cover a large area depending on its operating parameters. Furthermore, the coarseness of the information provided by radars is less prone to raise privacy concerns when compared to cameras.

The Housevibe system enables monitoring applications in which humans moving in a certain space have their overall activity quantified and transmitted to a remote location where it is logged for further analysis. Movement quantification takes the form of an index proportional to amount of a person's physical activity in the monitored area: the more movement, the higher the index value returned by the system.

The co-existence of on-body and off-body sensors increases usability, coverage and reliability of the monitoring system. For elderly care applications in particular, remote devices such as Housevibe address the problem of forgetfulness of users and non-compliance with wearable devices. Figure 2.5 shows the HouseVibe setup.

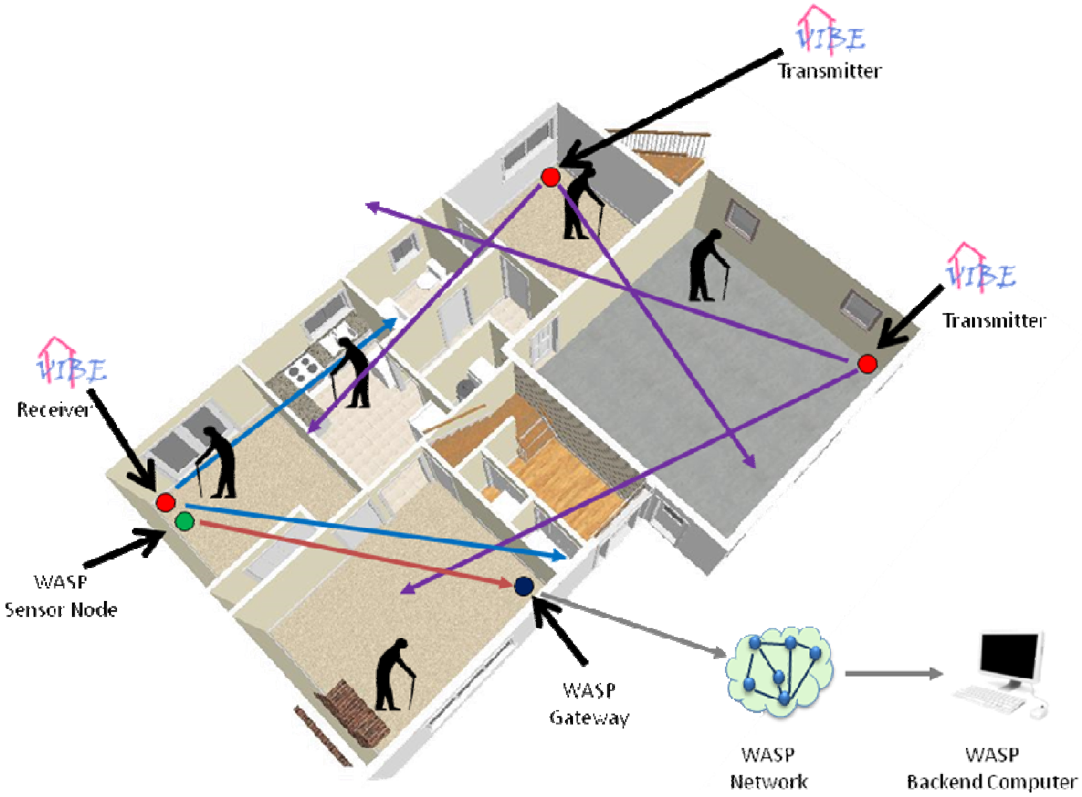


Figure 2.5 – Housevibe coverage of a house and combination with WASP nodes.

2.2.2.3 Behaviour profiling

The long term monitoring of activities could reveal alterations in lifestyle that might reflect changes in an elderly person's health. In this scenario, we will aim to collect data from a subject over a long period and correlate this information with information obtained from the person regarding his/her daily routine.

2.2.3 Hypothermia detection

Instead of measuring on-body temperature which could pose problems of wearability for elderly subjects, this scenario aims to monitor all rooms in a house and indicate if temperatures fall below a certain threshold. This threshold can be remotely set from the clinician's office. The system generates alerts that can be viewed on the backend, which can help a community matron locate elderly homes where there are temperature drops below acceptable levels.

2.2.4 Cardiac monitoring

In this scenario, arrhythmia detection will be implemented on the node allowing an alert to be sent to the backend. 90 sec of raw ECG data will then be sent to the backend in order to allow the clinicians to assess the arrhythmia further. Alternatively, if no arrhythmia is detected, an OK signal including the current patient's heart rate is sent to the backend at regular intervals. The Telmed system will be used to show alerts as well as raw data from the node (as shown in Figure 2.6)

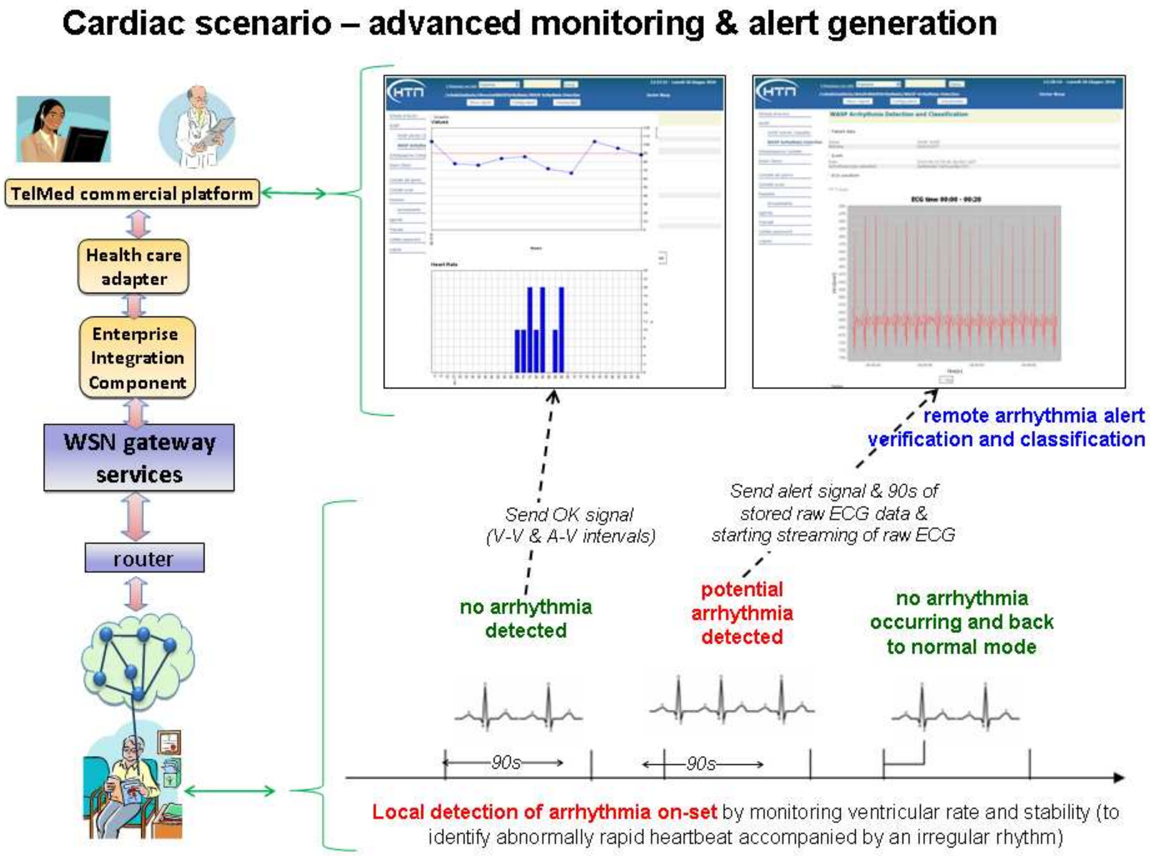


Figure 2.6 – Cardiac monitoring using the WASP system.

2.2.5 Fall detection

Falls are a serious problem in the aging population. They can indicate deterioration in health and are related to a decrease in musculoskeletal and motor-control function, with identified rates of mortality and morbidity in the elderly [2.7]. After the age of 60, there is a general rise in the number and severity of the falls, with an increased tendency linked to further complications. After this age, these rates increase substantially [2.8]. Falls affect elderly living not only in the community, but also in nursing homes and hospitals. For elderly living in the community, however, help or medical assistance can be difficult to summon and victims could remain without assistance for many hours.

While falling is not in itself a normal aspect of aging, it can be indicative of an underlying medical problem or some environmental hazard. These are defined as risk factors and can be classified as either intrinsic, or, extrinsic. Intrinsic risk factors generally include changes in the faller due to the aging process such as changes in cognitive, visual, or neurological functioning and musculoskeletal, sensory or cardiovascular impairment. Extrinsic factors are generally related to the environment including trip hazards, poor lighting, lack of hand rails, and some types of medication [9]. There are often complex interactions between the intrinsic and extrinsic risk factors that can increase the likelihood of a fall. For example, a faller with poor vision may not see a telephone cable in a poorly lit room and trip over it.

In the WASP project, we will use a NF(non-functional aspect) demonstrator to show the capability of the system to deal with falls. We aim to monitor the aftermath of a fall in order to infer whether the fall has no consequences or it requires the intervention of a doctor or a clinical specialist. More specifically, if after a fall the person wearing the node is able to walk, we can consider it as a fall without important consequences. If otherwise we don't detect any further activity after the fall, we can assume that the fall had compromised the ability of the subject wearing the node to stand and walk, and thus an alarm can be triggered.

Continuously calculating the activity of a node is a power consuming task. With the aim of improving the battery life of the node, we can perform such task only after a fall has been detected, using a simple and fast fall detection algorithm for such purpose. Using the NF framework allows to implement such scenario, as the NF manager can dynamically decide whether to use a simple fall detection algorithm or the more complex activity classifier.

By default, the NFmanager lets the simple algorithm run, feeding it with the data coming from the acceleration sensor on the node. When the simple algorithm detects a fall, the manager switches the execution to the activity classifier. Once the activity classifier starts its execution, the NF manager can decide to switch back to the simple fall detector when the activity class goes back to WALKING (which means that the person wearing the node has fully recovered from the fall). As long as the activity class remains fixed at a STANDING value, we are constantly interested in discovering whether the person wearing the node has recovered, therefore the manager continues the execution of the activity classification function.

2.2.6 Scalability

In this scenario, the ability of the WASP framework to deal with a large number of nodes will be investigated. The scenario will cater for an elderly home where subjects are highly mobile and will use different sensor modalities. This scenario will be detailed in deliverable 8.4 and will involve a simulation and study of network load and capability rather than a real-time demo.

2.3 Algorithm and software development

The scope of this Chapter is the description of the process used to implement several of the algorithms required for the EC scenarios. A specialised protocol stack (the BAN stack) was adapted for the EC test bed as well as a specialised programming model (ECA). Details of these components are available in deliverables of WP3 and 4.

2.3.1 On-node Classifier

The on-node classifier is designed to provide a means of calculating activity on the node instead of sending raw data. This would preserve resources in the network and allow for the use of a larger number of nodes if needed.

The on-node classifier was adapted from a paper by ICL [2.10] and written for use in WASP by WASP partners. The classifier is a Multivariate Gaussian Bayes classifier and can be implemented on the node as it is light-weighted in terms of computation but has an intrinsically efficient inference capabilities. As extensive memory is required for training the Bayes classifier, the training is conducted offline where multivariate Gaussian density is used to model the different activity classes. Under this framework, the likelihood of multivariate feature vector x (variance across 3 axes) belong to class Cj is defined as:

$$P(x \mid C_j) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|\Sigma_j|}} \exp\left[-\frac{1}{2} \left(x - \mu_j \right)^T \Sigma_j^{-1} \left(x - \mu_j \right) \right] \qquad (1)$$

where n is the dimension of the feature vector, and µj and ∑j denote the mean and covariance of class Cj respectively. The mean µj and covariance ∑j are derived from the training data.

$$\mu_j = \frac{1}{N_j} \sum_{x \in C_j} x$$

$$\sum{}_j = \frac{1}{N_j} \sum_{x \in C_j} x x^T - \mu_j \mu_j^T \qquad (2)$$

where Nj represents the number of features belonging to class Cj.

According to the Bayes theorem, the posterior probability of activity class j given feature vector x is:

$$P(C_j \mid x) = \alpha P(x \mid C_j) P(C_j) \qquad (3)$$

where α denotes the normalising constant and P(Cj) represents the prior probably for class Cj. To minimise the bias towards more frequent activities, activity classes are assumed to have equal prior probabilities. The classification can be simplified as follows:

$$d_j(x) = -\frac{1}{2} \ln |\Sigma_j| - \frac{1}{2} \left[\left(x - \mu_j \right)^T \Sigma_j^{-1} \left(x - \mu_j \right) \right] \qquad (4)$$

where x is belong to class Cj if dj(x) > di(x) for all j≠i.

To enhance the accuracy of the classification, the signal amplitude is used to identify static activities (such as lying down or sleeping) and falls, and the classifier is used to determine other dynamic activities (such as running and walking).

Offline data collected for training the classifier is shown in figure 2.7. This offline data is used to determine the parameters of the classifier and use it for on-node classification. These parameters can be changed if subjects of a different age-range or if impaired subjects are included.

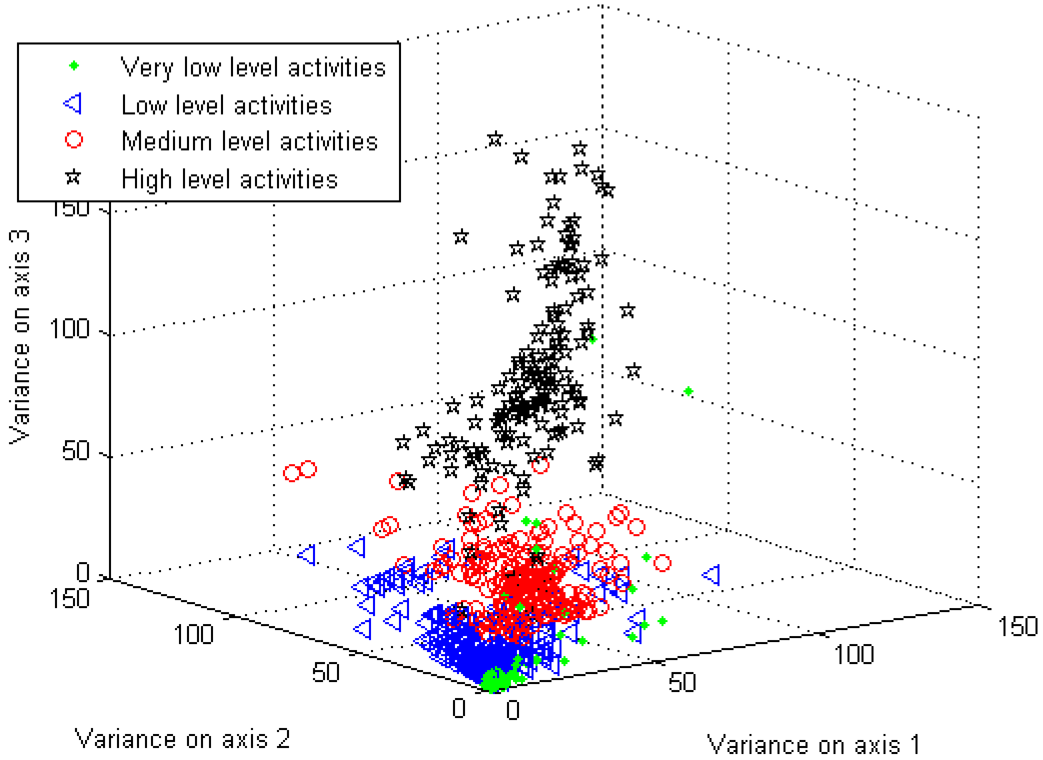


Figure 2.7 - Variance across 3 axes from the offline lab data.

2.3.2 Alarming

Alarming is used in several of the above mentioned scenarios. Two important cases are sending an alarm in case an arrhythmia is detected by on-node algorithms using ECG data, and that of alarming in case of the detection of hypothermia in one of the rooms. The advanced feature of using alarming in WASP is the ability to change thresholds from a distance, which means that medical staff could modify parameters on a patient's node from a distance without having to be physically present with the patient in order to access the node.

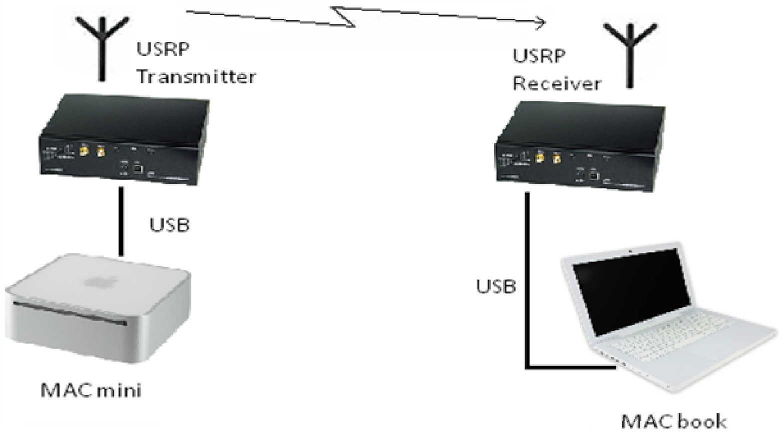
2.3.3 The use of ambient and wearable sensors

2.3.3.1 The house-vibe radar

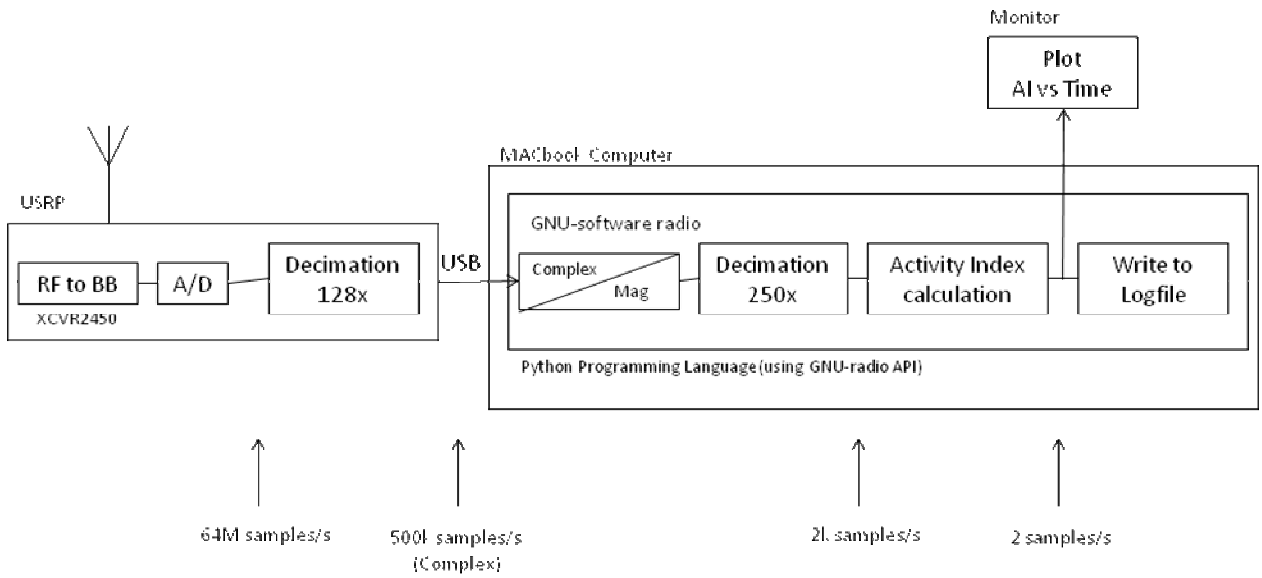
The HouseVibe radar is comprised of one or more radio transmitters continuously broadcasting an unmodulated wave and a receiver that estimates activity level in the environment based on variations in the strength of the received signal. A prototype of the radar was implemented using GNURadio, a free programming toolkit for realizing software defined radios and a pair of Universal Software Radio Peripherals (USRP), a general purpose programmable software-defined radio platform. Each USRP interfaces with an external computer responsible for part of the signal processing needed for transmission and reception of radar signals (*cf.*figure 2.8 (a)). The USRP is a digital acquisition (DAQ) system containing four 64 mega sample-per-second (MS/s) 12-bit analogue-to-digital converters and four 128 MS/s 14-bit digital-to-analogue converters. Several transmitter and receiver plug-in daughter boards are available covering various bands between 0 and 5.9 GHz. The current set-up uses the XCVR2450 daughter board, a 2.4 – 2.5 GHz and 4.9 to 5.85 GHz Dual-Band Transceiver (100+mW output on 2.4 GHz, 50+mW output on 5 GHz).

The transmitter, running on the MAC mini, is configured to transmit an un-modulated 5 GHz carrier. The Activity Index (AI) detector algorithm runs on the MAC book. The total receiver chain is shown in Figure 2.8(b). The decimation rates inside the USRP and in the software radio are configurable. By using the settings as shown, AIs are produced at 2 Hz rate (2 data points per second).

The Activity Index (AI) is calculated as the ratio between the standard deviation and the mean of the received signal strength over a sliding window of about 0.5 s duration. The AI data is written to a log file and to the screen *cf*. figure 2.9.



(a)



(b)

Figure 2.8 – Housevibe radar (a) and detecting activity indices (b)

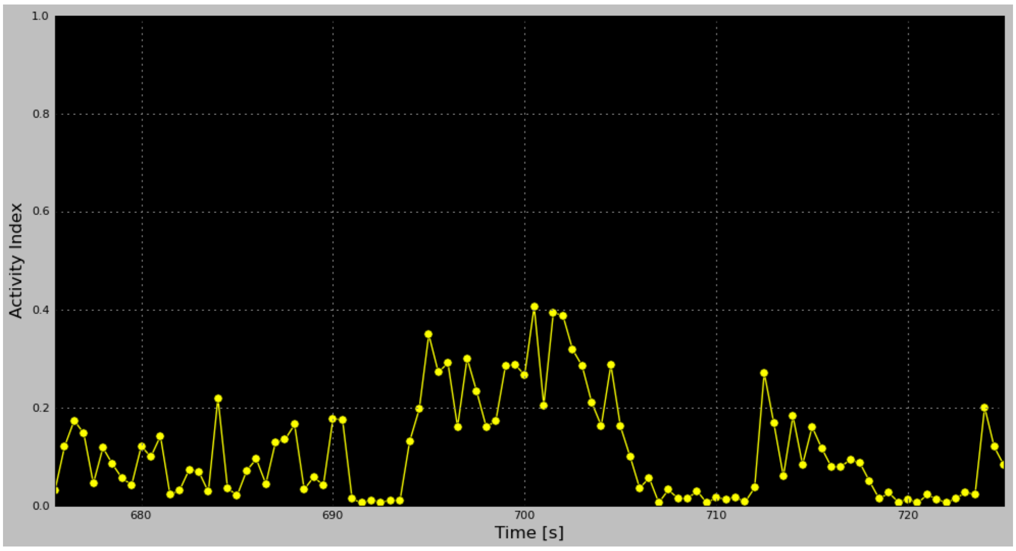


Figure 2.9 - Activity Index Plot- Vertical axis shows activity index and horizontal axis time in seconds. The higher the index, the more activity was sensed.

In order to make the scenario more complete, the interactions with mobile vehicles and maintenance operators are taken into account.

First of all, maintenance operators must be able to access the data of every node in order to check the status and the correct functioning of sensor devices.

Secondly, emergency vehicles (low-mobility) must be able to work as gateway in emergency situation and collect all the information available in the WSN, without interfering with the data exchange between the WSN and the central infrastructure.

In order to consider these features, one additional use case is here investigated:

UC4 - check of the sensors status and battery level by maintenance operators and service/emergency vehicles.

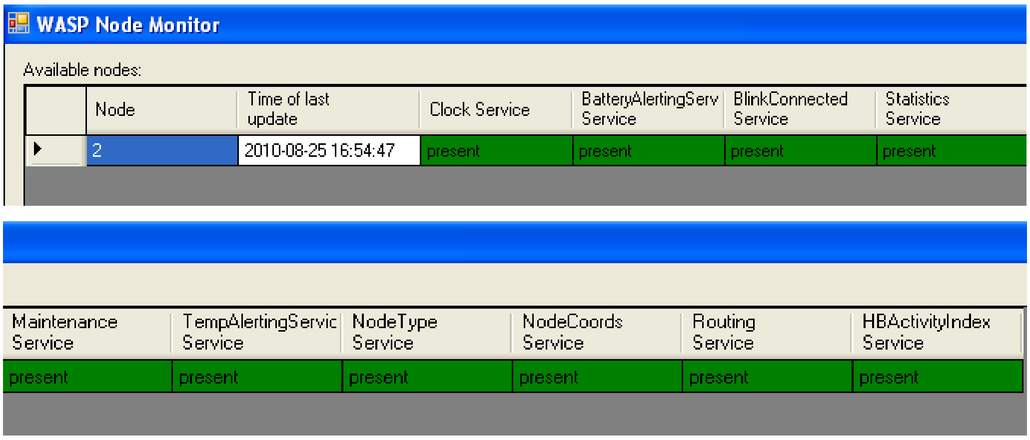
2.3.3.2 WASP node interface, gateway and visualiser

The radar interfaces with the WASP infrastructure via a USB serial port connected to a BSN Node, cf. figure 2.10 (a). A python script (AIinterfBSN3.py) periodically checks for a file (AIinterfBSN) containing a single AI data point. When found, it reads the AI value and deletes the file to ensure that the same data will not be read again. The AI detector software in the computer writes every half a second a new file containing a fresh AI. The sampling rate of the interface script it is set to 1Hz (1 data point per second).

The BSN Node interfacing with the Housevibe radar runs an application (housevibeEca.wsp) consisting of several services, cf figure 2.10 (b).Transmission of AI data points via the WASP wireless network is accomplished via the "HBActivityIndex" service. This service contains a function named "SendToSubscribers" responsible for transmitting data to the subscribers of the service. The function requires five parameters: '$handler'; 'NodeID'; 'seqnr'; 'HBActivityIndex'; and 'HBtStamp'. The '$handler' parameter indicates who actually process the data dispatch; ''NodeID' represents the identity of the reporting node; 'seqnr' is the sequence number of report; 'HBActivityIndex' is the calculated Activity Index; and 'HBtStamp' is a time stamp.

The WASP Elderly Care Activity Visualiser, *cf*. figure 2.10 (c), is the application subscribing to the elderly care ECA service. It displays the activity level coming from the Housevibe sensor node. Currently, this level is updated every second and it cannot be changed by the end user.

(a)



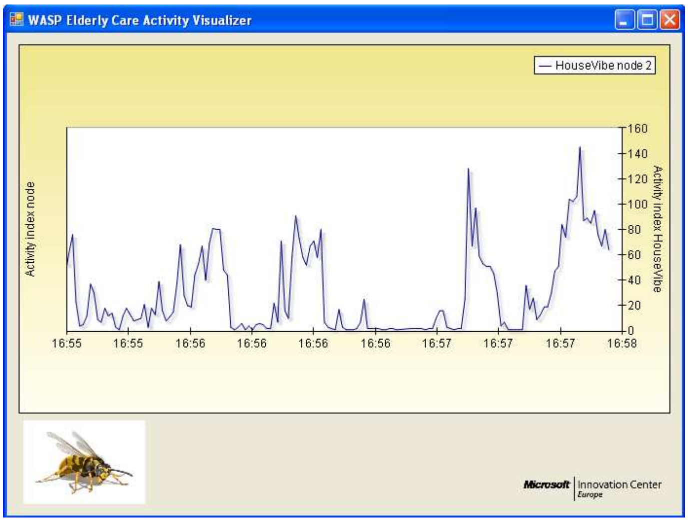
WASP Node Monitor

Available nodes:

	Node	Time of last update	Clock Service	BatteryAlertingServ Service	BlinkConnected Service	Statistics Service
▶	2	2010-08-25 16:54:47	present	present	present	present

Maintenance Service	TempAlertingServic Service	NodeType Service	NodeCoords Service	Routing Service	HBActivityIndex Service
present	present	present	present	present	present

(b)



(c)

Figure 2.10 - BSN Node attached to Housevibe radar receiver (a) ,house-vibe node monitor (b) and visualiser (c).

2.3.3.3 Evaluation and method

In order to verify the accuracy of the Housevibe system we conducted an experiment in which a person moves about a room according to a pre-defined script. The movement script indicates the activity level the person should exert at each interval of time. During intervals in which the script number is high, the person should run or jump, while low script numbers designate static positions or slow motion. The objective of the experiment is to check the agreement rate between the activity level indicated by Housevibe and the pre-defined script. Activity level data provided by on-body accelerometers were also included in the experiment as an extra source of movement information.

In order to compare the information provided by all the quantification methods (Housevibe, Accelerometer, Script) their data were categorized in one of three classes: HIGH, MEDIUM and LOW activity. Classification was accomplished according to the statistical distribution of the data provided by each quantification approach. Data inside the first standard deviation from the average was considered MEDIUM; while data outside this interval was consider HIGH and LOW depending whether their magnitude was above or below the mean. The percentage of matches between the activity classes of Housevibe and Script as well as Accelerometer and Script were then computed. A high match rate indicates the sensor has a good ability of categorizing human movements according to intensity.

2.3.3.3.1 Setup description and experiment

A person's movement in a confined area was measured using radar transmissions of 5 GHz and 30 dBm (including antenna gain). Figure 2.11 shows the setup of the testing environment.

The dimension of the room is 8x11 m^2. The USRP Transmitter connected to the mini MAC is on the desk right next to 'Circle 2', whereas the USRP Receiver and the BSN Sensor Node are situated on the desk next to 'Circle 4'. The 'Gateway' is located next to 'Circle 5'.

The on-body accelerometer sensor, an Actiwatch Spectrum Device produced by Philips Respironics[1], was attached on the subject's arm. Due to the limitations in the software of this device movement data is produced at intervals of 15 s only.

The pre-defined movement script used as activity baseline starts at 'Circle 1' where the subject is required to stand motionless for 15 s. The subject then must walk for 15 s towards 'Circle 2', where he will stop and wait for another 15 s. Next, he will move to 'Circle 3' for the following 15 s. After waiting for 15 s in this point, the subject will start jumping for the next 15 s. Subsequently, the subject will stand still for 15 s, after which he/she will move to 'Circle 1' again in another 15 s move. See top of figure 2.10 for a pictorial representation of the movement script.

This cycle will be repeated a total of five times (total time per cycle: 120 s, out of which 60 s standing and 60 s moving; Total time 10 minutes). All sensor data collected is averaged over 15s-interval values and plotted, as illustrated below (*cf.* figure 2.12). All data is analyzed using MATLAB.

[1] http://actiwatch.respironics.com/

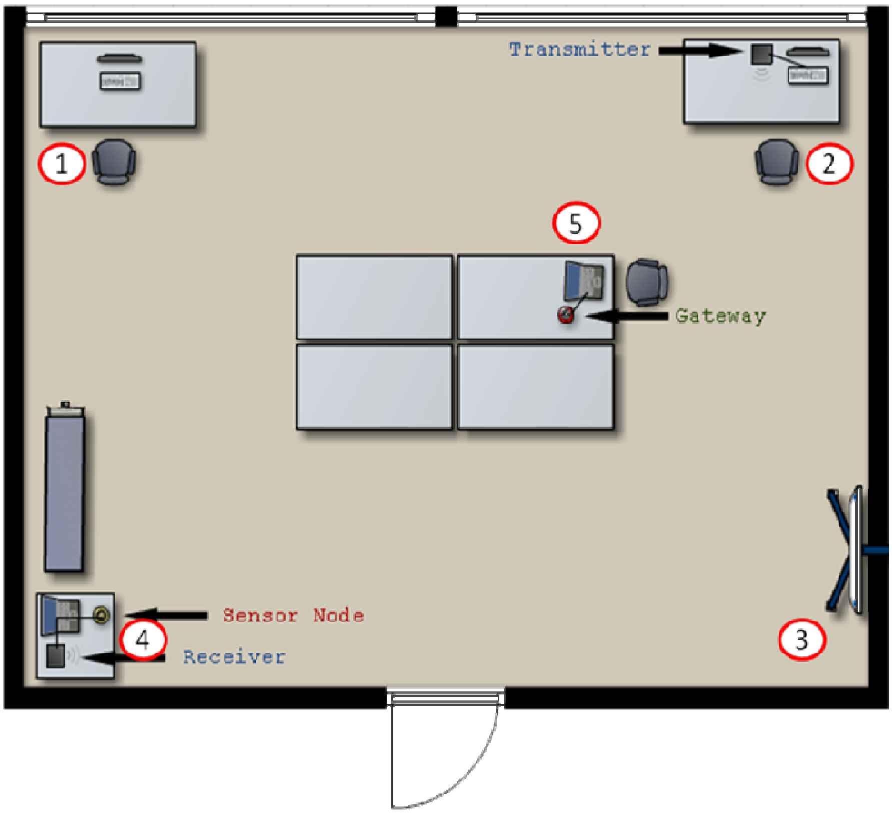


Figure 2.11 – Experimental setup

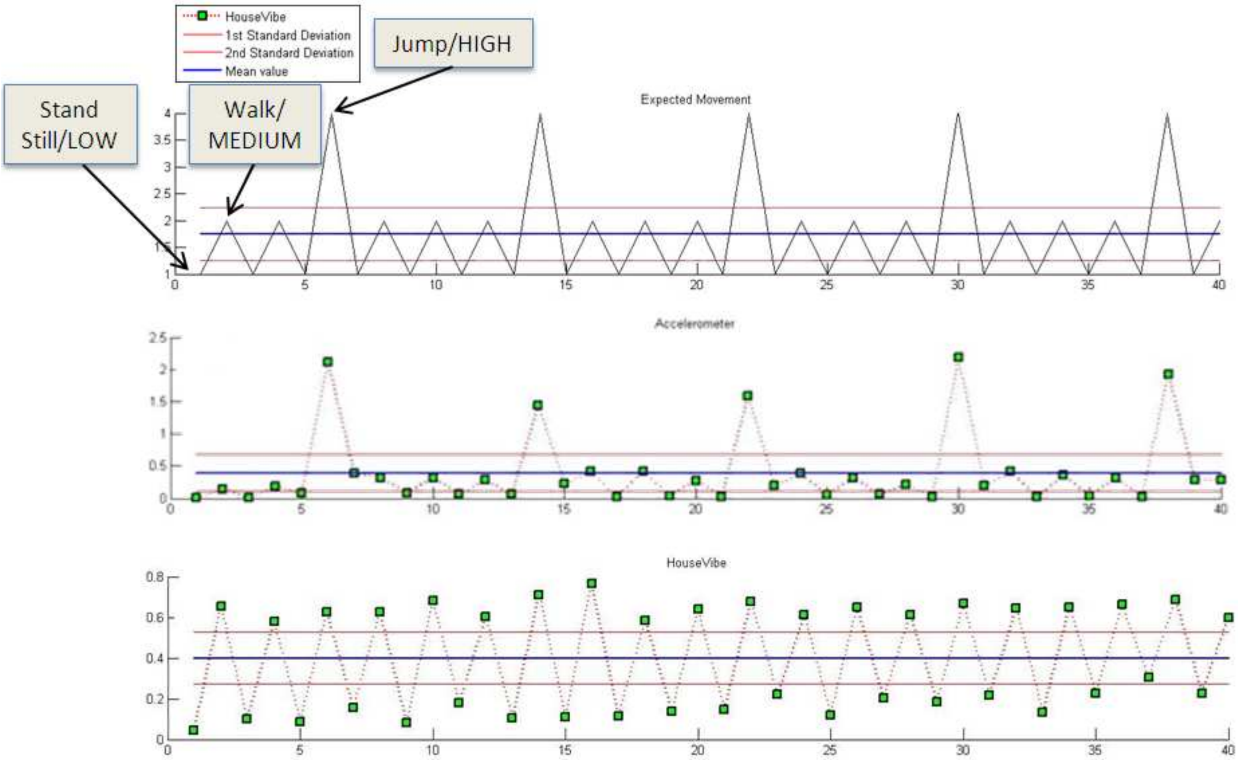


Figure 2.12 - Script Data and sensor measurements. Horizontal axis indicate time (unit=15 seconds) and vertical axis indicates numeric activity level (scale depends on the sensor/strategy)

2.3.3.3.2 Results

The matching rates of the classification of activity level according to the statistical distribution of the data are shown in Table 2.1. A high matching rate to the expected activity level indicates the sensor has a good ability of categorizing human movements according to intensity. The statistical significance of the matching rates was computed using the Wilcoxon signed-rank[2] and rank-sum test[3].

Comparison of the Housevibe data with the expected motion activity reveals a matching of 60 % and a correlation factor of 0.78. It shows us that the off-body sensor is able of accurately detecting high index values (i.e. jumping, running) and low index values (static movement). Housevibe is however unable to accurately identify intermediary activity levels. In general, MEDIUM activity is classified as HIGH by Housevibe (*cf.* figure 2.12). Therefore in its current implementation Housevibe is an accurate binary sensor of human movement (*i.e.* movement/no movement).

As expected, the on-body sensor performs better then the Housevibe with a matching of 85 % and a correlation factor of 0.92. A comparison between HouseVibe and Actiwatch indicates a matching factor of 47.5 % and a correlation of 0.55.

	HouseVibe vs Expected	Actiwatch vs Expected	HouseVibe vs Actiwatch
Points that correlate (percentage)	HIGH = 100 MEDIUM = 0 LOW = 95 Matching = 60	HIGH = 100 MEDIUM = 93 LOW = 75 Matching = 85	HIGH = 26 MEDIUM = 0 LOW = 74 Matching = 47.5
Correlation factor	0.78	0.92	0.55
Statistical significance	Signed rank = 3.6e-8 Rank-sum = 7.21e-15	Signed rank = 3.6e-8 Rank-sum = 7.21e-11	Signed rank = 0.016 Rank-sum = 0.02

Table 2.1- Experimental results

2.3.3.3.3 Summary and comments

In this section we have described HouseVibe, a radar-based monitoring system that provides information on human movement in a given indoor area, and the steps undertaken to add the radar as a node in a WASP network. We conclude this section with an outline of lessons learned during the integration process and list of feature recommendations for future generations of the WASP application toolkit from the point of view of an application developer.

[2] http://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test

[3] http://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U

Overall, the WASP application programming architecture provides an intuitive and flexible way to integrate sensors in a networked system. The virtual machine based programming model obviates the need for recompiling the entire code again in case there is a need of changing simple things such as the sampling period in which sensors report data. The language constructs for prototyping data collection applications were straightforward and offered no difficulty for programmers exposed to them for the first time. On the negative side, the application toolkit still lacks certain components that are important for prototyping and deploying real system. The most import missing tool in our integration exercise was a software module capable of easily allowing streaming data from a computer to a sensor node via the USB serial port. For this purpose, we, as application developers, had to access and modify lower level code. The lack of such tools is however natural and expected in a still maturing system such as WASP. Based on this experience we strongly recommend efforts directed towards making the system friendlier to the application programmer. This effort may take the form of Integrated Development Environments (IDE), better support for a wide range of operating system (e.g., OSX) and the missing libraries and components for expected needs (e.g. data stream through USB serial port).

2.3.4 On-node ECG monitoring

The on-node ECG monitoring provides a way to detect and classify arrhythmic events on the node.

The automatic arrhythmia detection and classification process is based upon:

- a *sampling* process of raw ECG signal coming from a sensor board attached to the node
- *storage* of the raw sample on the node memory
- *cancelation* and *noise removal* for raw ECG signal applying a set of filters to each sample
- *QRS complex extraction* process based on *Engelese and Zeelenberg algorithm* [2.11] (the QRS complex is a recording of a single heartbeat on an ECG that corresponds to the depolarization of the right and left ventricles)
- *three-steps arrhythmia detection and classification* process based on the *ELA Medical PARAD+ algorithm* [2.12]: arrhythmia detection triggers the alarm mode
- *heart rate calculation* and *transmission of data*: full ECG signal when in alarm mode, a control message containing last calculated heart rate if no adverse events are detected

The on-node ECG monitoring process was implemented using the uDSSP programming model and according to the subscription-based interaction pattern used in the WASP project.

As shown a clinician can start the monitoring process of a patient from his health record in the TelMed platform. His request will go through the whole WASP stack, from the HCA to the EIC till the node, where the local monitoring process will start: in case of an adverse event, the node will start to transmit 90 seconds of the full raw ECG signal to the upper layers.

More in detail the full process consists of the following steps:

- **service subscription:** as soon as a business application subscribes to the patient monitoring process through the HCA and the EIC, the on-node processing starts locally
- **initialisation of service:** the node starts sampling the ECG signal with a sampling rate of 200Hz and converts it by an ADC (Analog to Digital Converter) with a resolution of 10 bits
- **storage in node's memory:** the raw sample is stored in node's memory to be sent later in case an adverse event is detected
- **signal filtering:** cancelation and noise removal are achieved applying a set of filters to each sample. The sample goes through a differentiator managed by a notch filter and then it passes through a cascade of two different FIR filters (a 16-order high-pass filter

with pass-band upper frequency of 2 Hz and a 16-order low-pass filter with stop-band lower frequency of 30 Hz). Filtered samples are then added to a buffer

- **QRS research:** after there are at least 40 samples processed by the low pass filter, the content of the buffer is evaluated against an adaptive threshold in order to extract a QRS candidate for the search region: if the amplitude is greater than the threshold we have found a QRS candidate. If additional threshold crossings occur, the occurrence is classified as noise, otherwise another conditional test is needed to classify the first crossing as a QRS
- **threshold adaptation:** the threshold value is dynamically changed, calculated using the root mean square of the last 256 samples
- **heart rate calculation:** once we have four measures we are able to calculate the heart rate using the sampling rate and the last fourth difference of the last one QRS and the previous QRS. If there is any active service subscriber, the new heart rate value is signalled by the node
- **majority rhythm classification:** the distance between two succeeding *R-waves* in the QRS complex is called *V-V cycle*: V-V cycles are stored in another buffer and after collecting eight of them we are able to group and classify them in order to identify what is called the *majority rhythm*
- **arrhythmia detection and classification:** once a particular majority rhythm type is identified (the *VT* one, *ventricular tachycardia* or the *VF* one, *ventricular fibrillation*), the arrhythmia detection event is raised and the process goes through further analysis of collected data, in order to classify *the type of arrhythmia*: for some types of arrhythmias, the classification process requires the calculation and classification of *A-V association* (stability of atria ventricular intervals). In order to obtain this information, we have to extract the *P wave* from the ECG signal and to perform complex analysis on data, but unfortunately this led to problems due to computational and memory constraints of the nodes: therefore complete arrhythmia classification takes place at the HCA level along with the detection of false positives
- **alarm mode:** arrhythmia detection triggers the alarm mode, in which the system suspends data analysis and the detection process, switching to the transmission of the full ECG signal (split in chunks) to the higher levels of the WASP system
- **normal mode:** otherwise in case no adverse events are detected, a control message including patient's heart rate will be sent to the higher levels of the WASP system

Further details on the implementation and the results for the ECG monitoring service could be found in Deliverable D5.7.

2.3.5 User-interfaces

An important aspect of the WASP project is the ability to allow users to observe different aspects of the system which are of relevance during both installation and deployment. Several user-interfaces were provided to observe the system, these include:

- **The topology monitor** to observe the topology as nodes are mobile.
- **A statistics monitor** to look at network statistics.
- **A visualiser** at the gateway level to observe the incoming data (shown in figure 2.9(c)).

These interfaces are used mainly during programming the nodes and testing them in order to make sure the system is running reliably. They can be checked later on to make sure reliability is maintained and to observe nodes that have rebooted.

2.3.5.1 The EIC interface

The EIC interface was introduced in detail in D6.3. The interface is used both for the Herd health and Elderly care test beds. For the EC test bed several modifications have been implemented to allow the interface to be used for data collection over long periods of time, these include:

- **Subscribing to different services:** This allows the display of data from different sensing modalities. The user can shift between temperature and activity for example. With this functionality, the user is able to subscribe to sensor data that is of relevance to his application.
- **Exporting data:** This can be over one parameter or all data received. Exported data is in a .csv file which can easily be used in other programs such as Matlab, for data analysis.
- **Displaying data:** Sensor data is displayed in the EIC interface only if the user subscribes to it. Depending on user's preference, battery, statistics over sensor data value (e.g., maximum, minimum, average) can be displayed.
- **Sending commands to sensor nodes:** A set of commands are available on the nodes in order, for example, to change the period of battery, sensor data sampling, or set a threshold on sensor data value alerting. In the EIC interface, the user can remotely send commands to the specific nodes, or to all the available nodes. This functionality is of extreme importance when the user does not have physical access to the sensor nodes. Figure 2.13 shows an example of using the EIC to send a command to the nodes.
- **Alarm display:** Alerts can be of two types: on-node alerts or alerts that are raised on the backend. Using the EIC interface, alerts can be send to the backend where they are displayed (for temperature-drops for example) the user can also define his own alert rules in the EIC interface. Figure 2.14 shows an example of defining alerts on the backend.

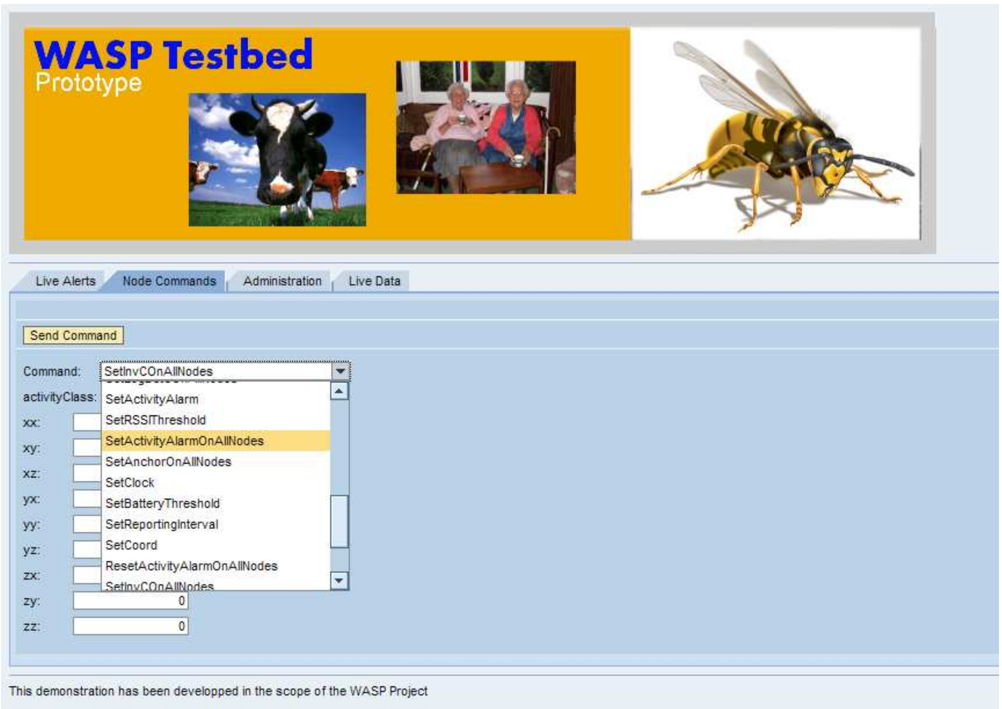


Figure 2.13 – Remotely sending a command from the backend to the nodes.

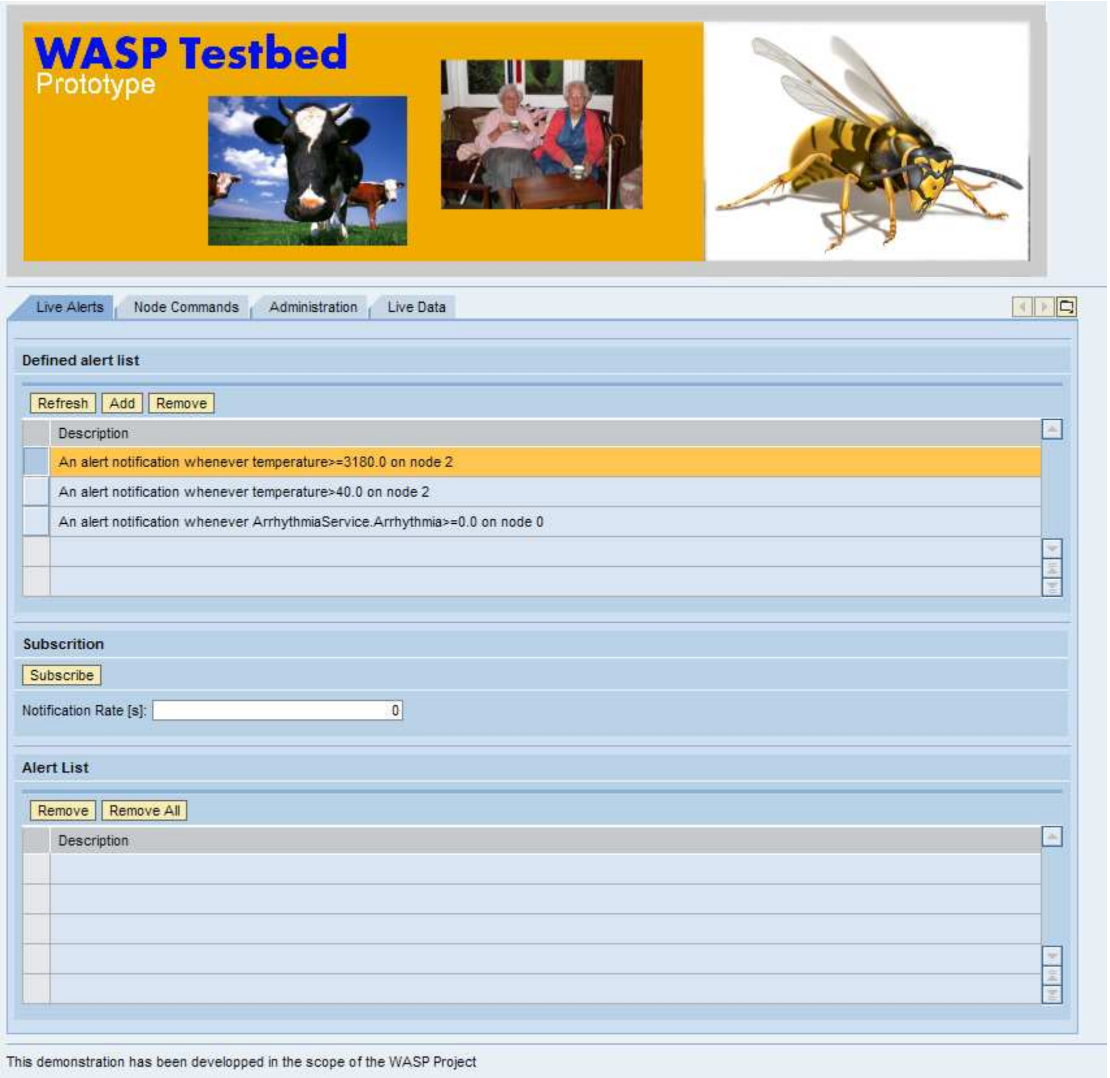


Figure 2.14 – Defining alerts on the backend relating to a drop in temperature (note is the display is as a voltage but can be changed to Celcius/Fahrenheit) or a detection of an arrhythmia.

2.3.5.2 The TelMed interface

The TelMed Platform was integrated for the final EC testbed as an example of real world business application using the services exposed by the WASP system.

Cardiologists can use HTN's platform user interface to start the monitoring process of a patient, and see data received from nodes, directly from the patient's electronic health record.

Figure 2.15 shows an example of the page where the clinician can set up the desired threshold for received heart rate values and the minimum notification rate.

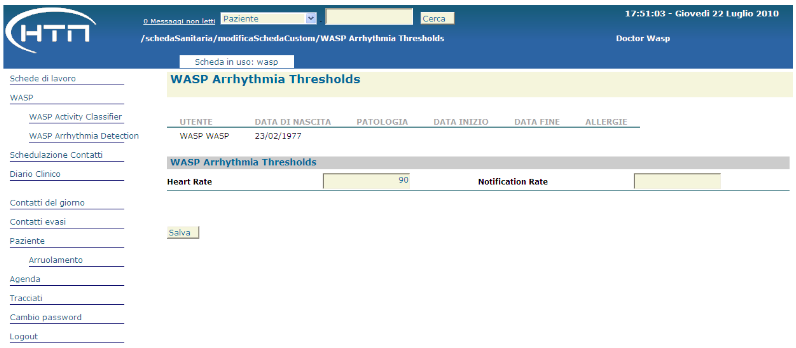


Figure 2.15 - Threshold values and notification rate setting in TelMed Platform

Once thresholds and notification rate are set, the clinician can start the monitoring process just clicking the subscribe button on the patient health record page: the request will be sent to the HCA and will then follow the whole WASP stack to the node used by the patient. At any time the monitoring process could be stopped in the same way.

Data received from the node are then stored inside the TelMed Platform database, and displayed in the patient health record. Figure 2.16 shows data received from the node for patient's heart rate value: data can be plotted in graphs and used for statistics, and may be filtered by date. Also, this report shows a list of arrhythmic events occurred during the selected period, which leads to another detail page where the single arrhythmic event can be inspected for further analysis.

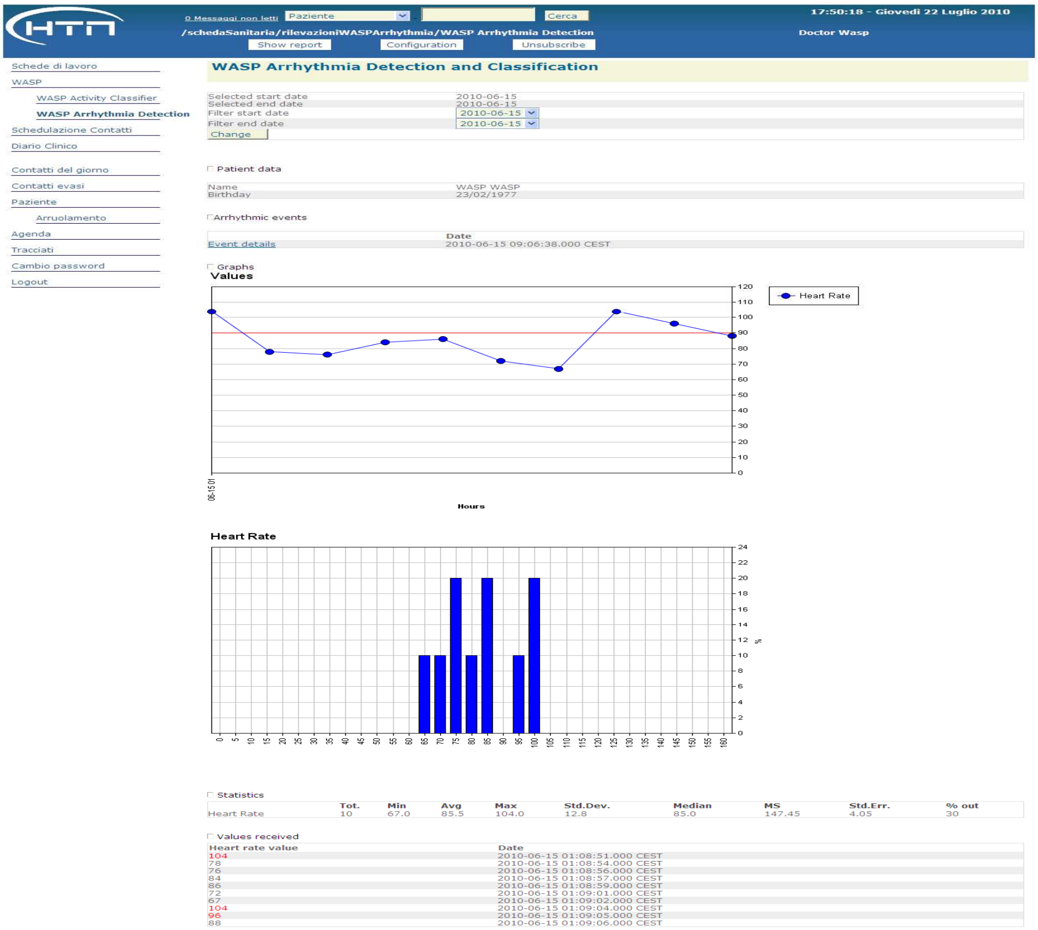


Figure 2.16 - Patient report for the ECG monitoring process in TelMed Platform

Figure 2.17 shows the detailed report for an arrhythmic event: from this page it is possible to see the details for the event and to inspect the full ECG signal received, and also confirm the diagnosis or insert notes which will be saved in the patient record system.

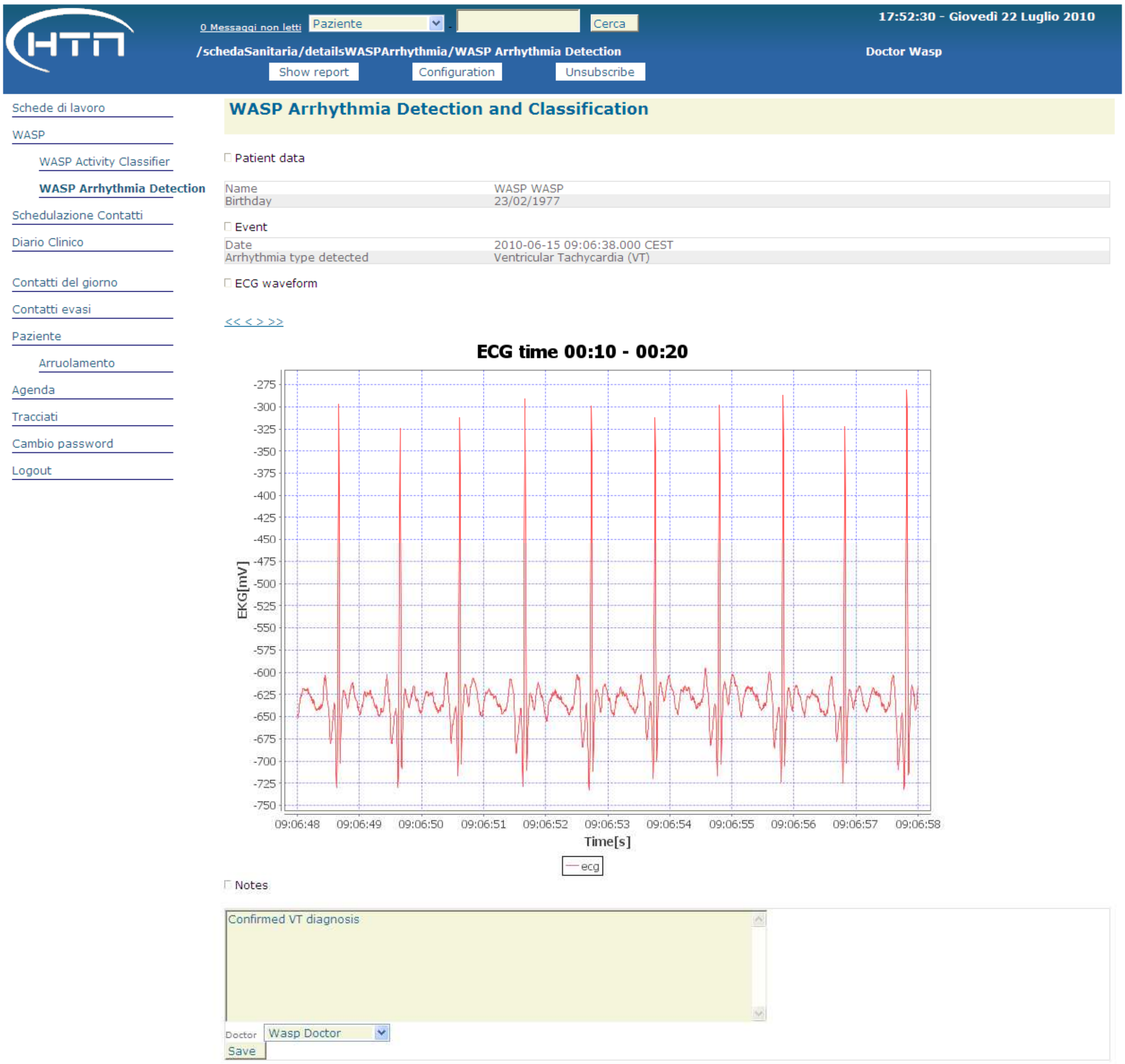


Figure 2.17 Arrhythmic event analysis and diagnosis in TelMed Platform

As described in details in Deliverable D6.5, the TelMed Platform exchanges data with the WASP system through a web service interface with the HCA. Subscription requests are generated from the patient health record page and sent to the HCA; in the same way ECG data sent from the node are recollected by the HCA and then sent to the TelMed Platform through a web service communication.

For the ECG service scenario, the HL7 Annotated Electrocardiogram [2.13] (*HL7 aECG*) was selected as a reference medical record data format: the aECG standard allows to submit both the ECG waveforms and the associated measurement locations (ECG annotations) in a standardized way.

The aECG standard was created by HL7's Regulated Clinical Research Information Management [2.14] (*RCRIM*). It passed final balloting in January, 2004, and was accepted by ANSI May, 2004, being reaffirmed in May, 2009.

Further details on the integration process of the TelMed Platform and on the format of data exchanged could be found in Deliverable D5.7.

2.3.5.3 The Maintenance interface

The maintenance interface allows the observation of people who are wearing sensors, their motion, activity levels and activity classification, as well as battery levels per node. It provides a visual interface which clinicians can use to observe the functioning of the whole system in relation to the patients (Figure 2.18).

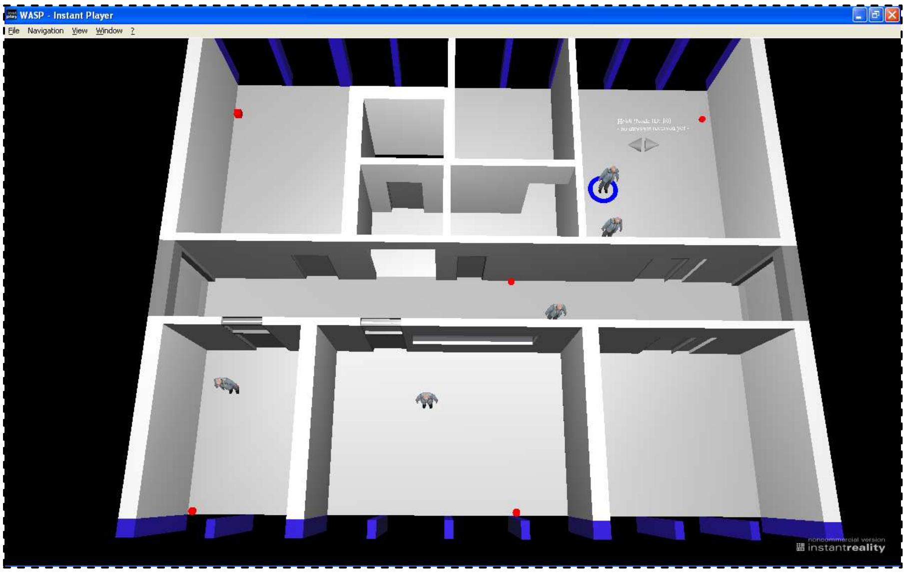


Figure 2.18 – The maintenance GUI allowing the observation of patients in the test bed.

2.4 Discussion and lessons learned

The integration proceeded as follows: partners were asked to develop their components separately then were asked to integrate them together and aim for a final demo in the ICL test bed. Many partners also borrowed nodes from ICL in order to implement and test their algorithms. The challenges encountered during the integration can be summarised as follows:

- **Ease of use of system:** ICL had a visiting researcher during the summer of 2010 working on WASP integration. The researcher was given the task of starting from scratch and installing WASP programs on nodes as well as testing the components leading to the backend. The WASP wiki was very helpful for obtaining directions so were the WASP partners who showed a good team effort to guide the researcher. The WASP cookbook was also used to obtain more specific information regarding components. The use of the SVN repository was fairly straight forward and allowed continuing update of software. However, some input was needed from expert partners in order to be able to use different components.
- **Reliability issues and difficulty of debugging:** One problem with the use of the system was that debugging was difficult for embedded code which made it challenging to identify bugs and parameter settings that causing delays in the system as well as rebooting of nodes. However, support from partners meant that many errors were ironed out in time. Once the system was up and running, debugging was not needed and the demos were implemented. The activity classifier was implemented by ICL but needed more support from partners to iron out issues that came up from implementing it in fixed C and then deploying it on the nodes.
- **Wearability of nodes**: Due to their size, the nodes were easily wearable (Figure 2.19). However, if used in healthcare monitoring, issues of ergonomics and bio-compatibility need to

be investigated. Nodes could be used in a plastic case that can be used on the waist, the arm, or the wrist (as a watch). Integration with the e-AR (ear worn activity recognition) sensor is also possible.

- **Flexibility in programming models**: Two programming models were available foe integration at ICL: ECA (provided by TUe) and uDSSP provided by EMIC. The flexibility allowed us to choose which model is appropriate for applications. For example, nodes can be programmed using the ECA model for activity/falls/alerts applications and with uDSSP for ECG monitoring applications. Reprogramming nodes was fairly straight forward.
- **Usability in patient trials**: Although discussions were on-going with clinicians who collaborate with ICL and HTN on the development of algorithm, more discussions are required if we are aiming to roll out this project for a large number of patients. The project has provided a set of very useful techniques for home-healthcare. However, to translate these techniques into patient studies requires selecting some of them (according to the study selected, such as activity monitoring of post-operative patients) and focusing on using those techniques for the specified study.

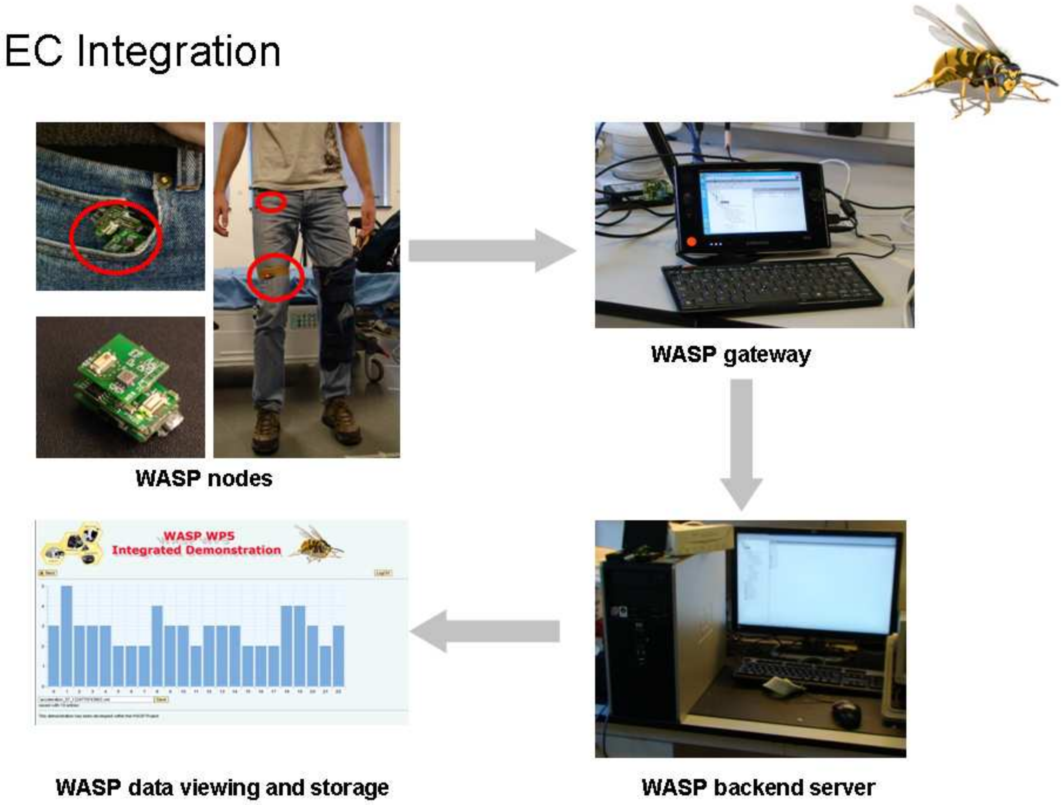


Figure 2.2 – Wearability of WASP nodes in the test bed.

3 Herd Control Prototype

3.1 Prototype description

In this chapter the prototype of ‘Detection of health problems with focus on claw health and locomotion’ is described in more detail. In the daily management the dairy farmer and his staff are responsible for the production level, the health and the well being of a cow. Important production related problems on dairy farms are mastitis (udder inflammation, clinical or sub-clinical), lameness and metabolic diseases. Timely diagnosis, if possible in the pre-clinical stage is important. From these problems lameness is chosen, since lameness and other locomotion problems in dairy cows often occur in current dairy housing systems. Locomotion problems also deserve more attention from the animal welfare point of view.

Locomotion is an important element of the cow's vitality. Healthy, free living cows move around in their living environments, often in species specific ways and at various gaits and speeds. Locomotion supports all main behavioural functions in that it enables the cow to act properly, in space and time, serving its different needs.

All locomotion and behaviour aspects are based upon measurements that are performed with the 3D accelerator sensor that is integrated in the WASP BSN node (Figure 3.1). The measurements that are performed are adaptive to the behaviour of the animal. The interval, duration and the frequency of the measurement are adaptive. In Figure 3.2 an overview is given of the different modes that are used during the 2010 HC Test Bed Meetings and Figure 3.3 gives an overview of what a cow is actually doing when a certain mode is activated.

The WASP recording system will monitor the animal in such a way that information comes available about animal’s location, activity, behaviour and gait. The basic part of the WASP system for this scenario consists of a 3-D accelerometer attached to the leg of the cow (Figure 3.1). In this way 3-D accelerometer sensors should be used for monitoring movements (activity and gait) as well as the position (lying vs. standing) of the extremities.

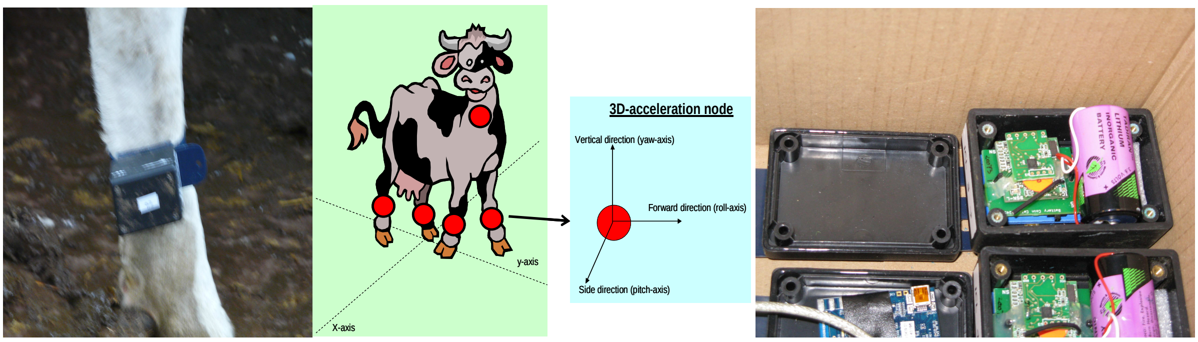


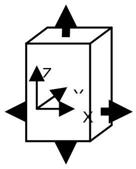
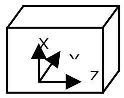
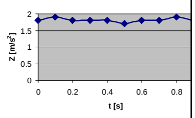
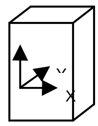
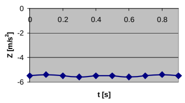
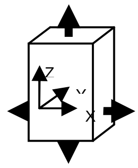
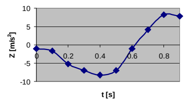
Figure 3.1- BSN nodes with sensors attached at the cow’s extremities for measuring location and 3D- acceleration.

Description of recorded parameters

- **Location** = position of animal in the horizontal x, y-coordinates in the barn or pasture

Reference nodes send position messages. Based on the strength of the position message (or time difference of arrival or angle of arrival) the relative position of the node is determined in relation to the reference nodes. From information on the absolute position of the reference nodes, the absolute position of the node is determined and stored together with a time stamp.

- **Locomotion** = movement and acceleration pattern of the claw in forward, sideward and upward direction during one step

	MODE Ø	MODE I	MODE II	MODE III
Behaviour	**-**	**Lying**	**Standing**	**Walking**
Interval:	Start/stop	1 per 2 min.	1 per 10 s	1 per 2s
Duration:	continues	1 s	1 s	1 s
Frequency	10 or 50 Hz*	10 Hz	10 Hz	10 Hz
		 	 	 
Values (raw):	30 or 150 per s	30	30	30
Values (aggregated):	2 bits [00]	2 bits [01]	2 bits [10]	2 bits [11]

* Data (3D) will be harvested with 10 Hz and during a limited period also with 50 Hz.

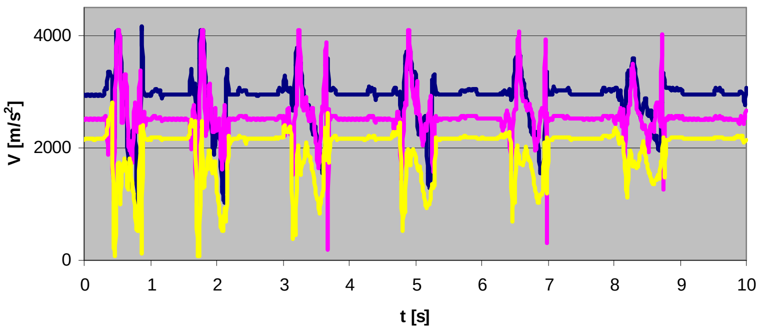
	MODE IV
Behaviour	**Locomotion**
Interval:	Interval: 1 / hour
Duration:	Duration: 10 s (valid when ≥ 5 steps detected)
Frequency:	50 Hz
	
Values (raw):	1500
Values (aggregated):	4 (Step duration, length, swing (amplitude of the sideward movement) and amplitude)

Figure 3.2 - Overview of the different WASP MODES used during 2010 HC Test Bed Meeting.

High frequency (50 Hz) recording of these values will provide a description of the track a claw makes during one step. Analysis of this signal should provide a number of parameters that give an assessment of the quality of the gait. Short-term changes in the locomotion parameters (locomotion pattern) give information about the gait of the animal when walking on different surfaces in barn or pasture. Long-term changes in the locomotion parameters indicate claw or leg problems.

- **Activity** = measuring value related to the number of movements or the displacement in the horizontal plane

From the acceleration values in forward direction the track a cow has travelled in a certain time frame can be calculated. Activity of an animal is generally lower when she has a claw problem or a metabolic disease. In case of an oestrus activity is increased. Activity will also increase when an animal has to put more effort in finding feed or water. The activity pattern of an animal compared with the mean herd activity pattern gives an indication of the rank of the animal. For example when the number of feeding places is restricted, low ranking animals will get a delayed access after fresh feed has been delivered.

- **Behaviour** = information about what the cow is doing at a certain moment; this can be standing, walking, eating, lying and is determined by angle of leg/head with tilt sensors (pitch, roll, yaw) or accelerometers.

When an accelerometer is oriented so that two axes are parallel to the earth's surface it can be used as a three-axis tilt sensor with a role, pitch and yaw axis. These recordings can be used to analyse what the cow is doing on a certain moment. This can be standing, walking, eating or lying. Recording behaviour during longer periods results in a daily behaviour pattern. From this daily lying duration, number of lying periods, lengths of lying periods can be calculated. This information can be used to assess the animal welfare status in relation with the housing system.

MODE I	MODE II	MODE III + IV
		

Figure 3.3 - Relation between MODE and what the cow is doing.

In the normal procedure the MODE 0 will not be activated. The MODE 0 is implemented because it is a nice tool for a system integrator for checking the WASP node and a farmer might want to have the possibility to start a continuous measurement for a certain cow when facing this cow.

There can be certain moments on a day where (almost) all animals will start walking, e.g. when fresh roughage is delivered at the feeding gate or when the cows are fetched for milking. During these moments it is likely that on almost all nodes the MODE 4 measurement will be activated. In big herds this could lead to the situation that during these events the radio link has not enough capacity. A trade-off mechanism regulating the radio use could in these circumstances be useful.

With regard to the user interface the original used case is depicted in Figure 3.4. The HC prototype focused on some parts that support the farmer. In the test bed the researchers of Wageningen UR were the users. Therefore the WASP HC prototype is not integrated in other management tools of the experimental farm.

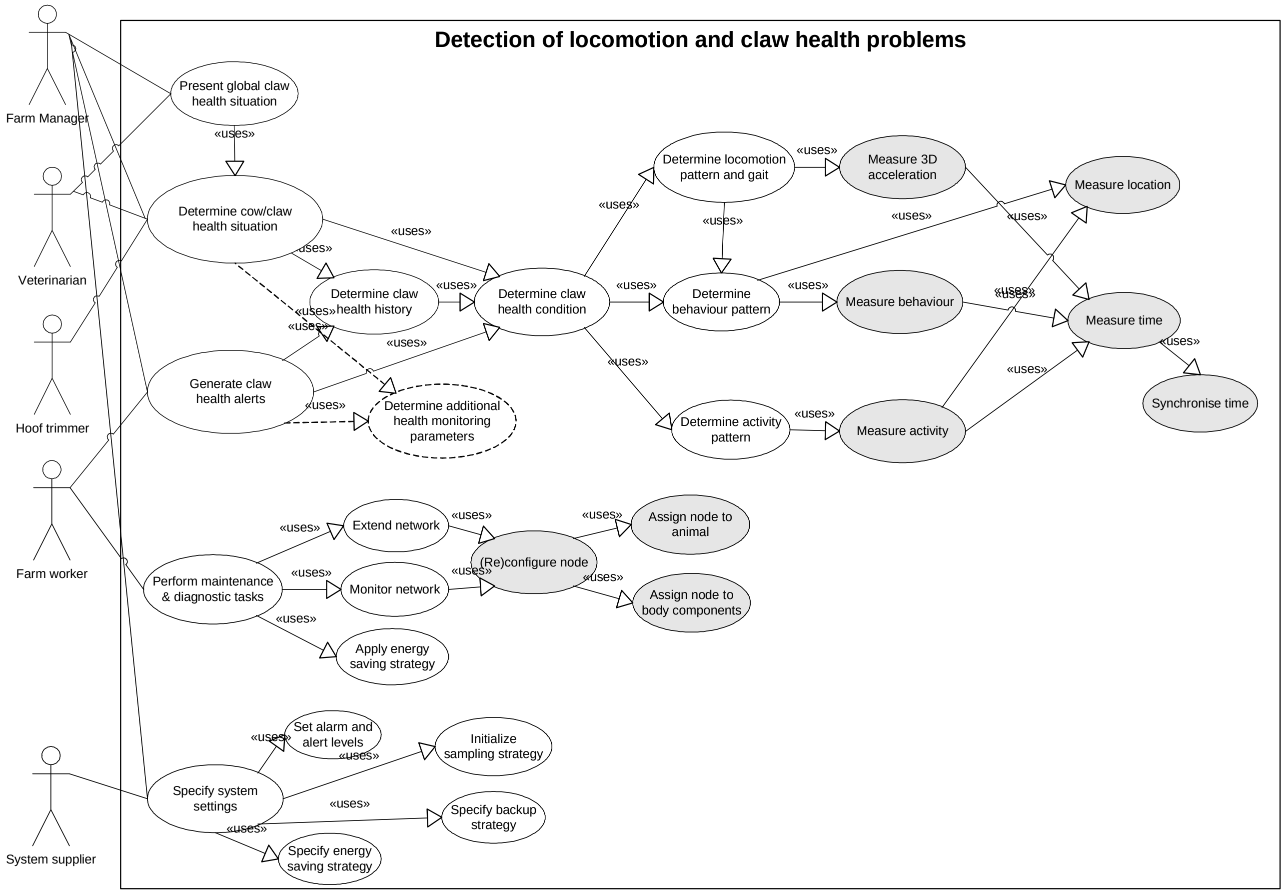


Figure 3.4 - Use case diagram of the scenario on "Detection of health problems with focus on claw health and locomotion"; use cases in grey are on the node level.

In Figure 3.5 an example of the user interface of the EIC (Enterprise Integration Component) of the WASP HC prototype and the overall picture are shown. The user interface is developed for system integrators. It is in general applicable for the EC and the HC scenario, since the connection to the nodes and the network by using the services ecosystem and the ECA and uDSSP programming models form WP5 form the basis. Of course there are some application specific services needed in the user interface.

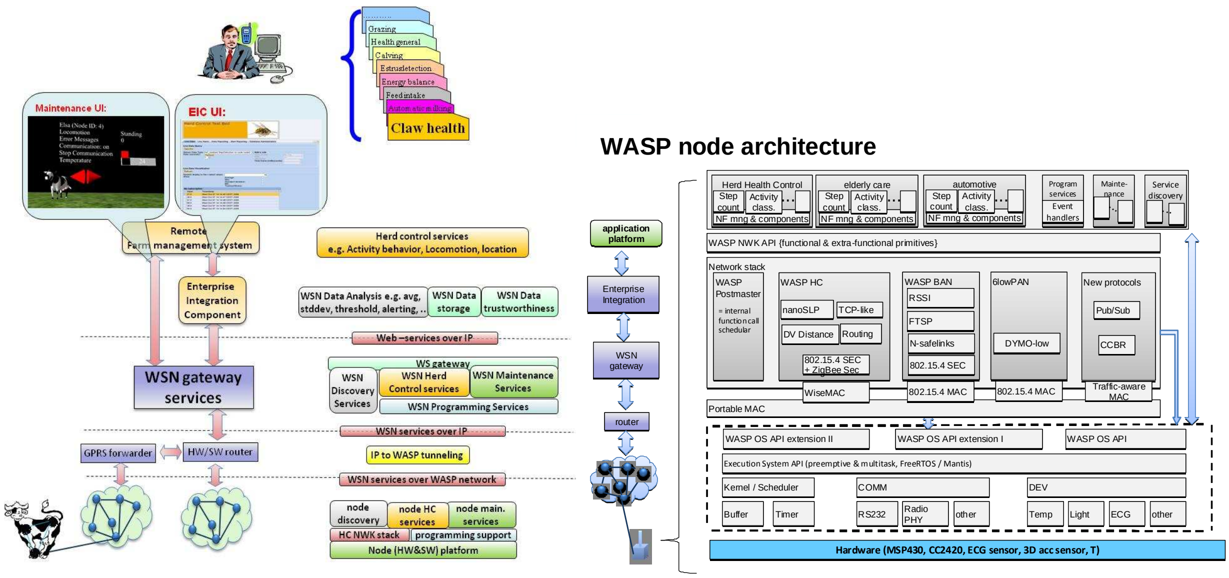


Figure 3.5 - Components View for HC Test bed

Development and testing of the HC prototype was done in an iterative process. In each iteration complexity, size of network and integration of WASP components differed. In the test results more information will be given on the specific set ups of the prototype. The original schedule of the testing periods is shown in Figure 3.6.

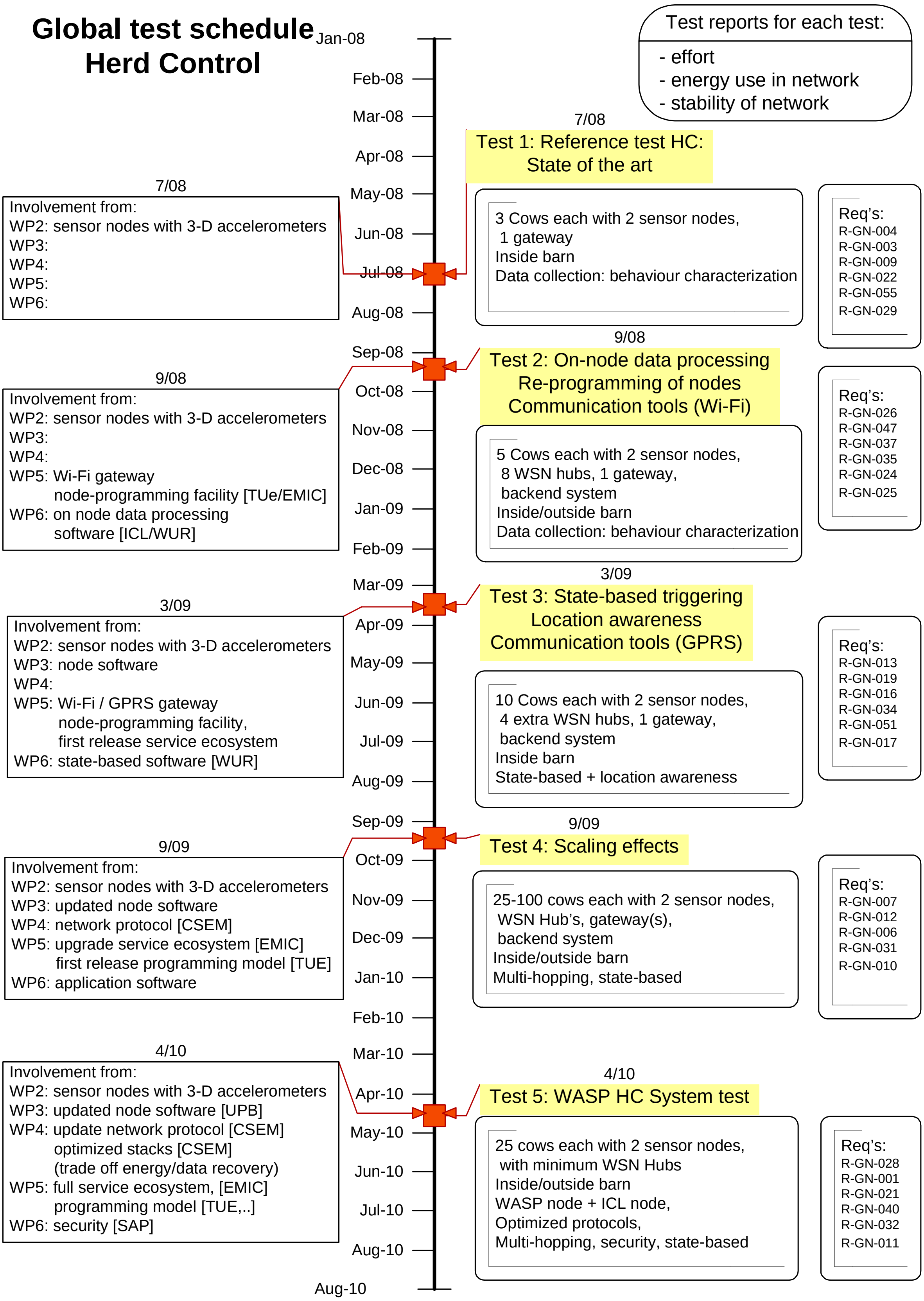


Figure 3.6 - Global test schedule for Herd Control test bed

3.2 Prototype development and testing

In D6.6 the intermediate results of the iterative process were presented as improvements made during each iterative step. At the end of the WASP project we would like to highlight the specific issues that were involved in the development and testing of the Herd Control prototype and the use of the test bed. In the following paragraphs the most important HC related topics will be discussed.

3.2.1 Hardware experience

The use of the hardware in combination with livestock is complex.

- The first problem that has to be solved is how to attach a sensor to an animal without causing injuries or discomfort for the animal.
- The environment is aggressive (moisture, ammonia, mechanical impacts), so the housing must be of high quality.
- The application of a sensor to an animal can be risk full (in relation to safety) and the catching of the animal and application of the sensor might be time consuming

The software used during the WASP project was not yet fully optimized for energy efficiency. To be able to have nodes active for a longer period big sized, high capacity batteries had to be used.

The ICL BSN nodes (Figure 3.7) used in the WASP project are developed for demonstration and education purposes. During the development of these nodes the robustness was not an issue. In the HC application the robustness is because of the aspects mentioned earlier and important aspect.

Body tissue has a big attenuating effect on high frequency signals as the 2.4 GHz radio signal used in the ICL BSN. Due to this effect it is in many cases impossible to receive a signal from a node if an animal is laying on this node.

There are differences between the 3D signals received from the BSN. This difference is caused by tolerance of the 3D sensors and small differences in the hardware configuration.

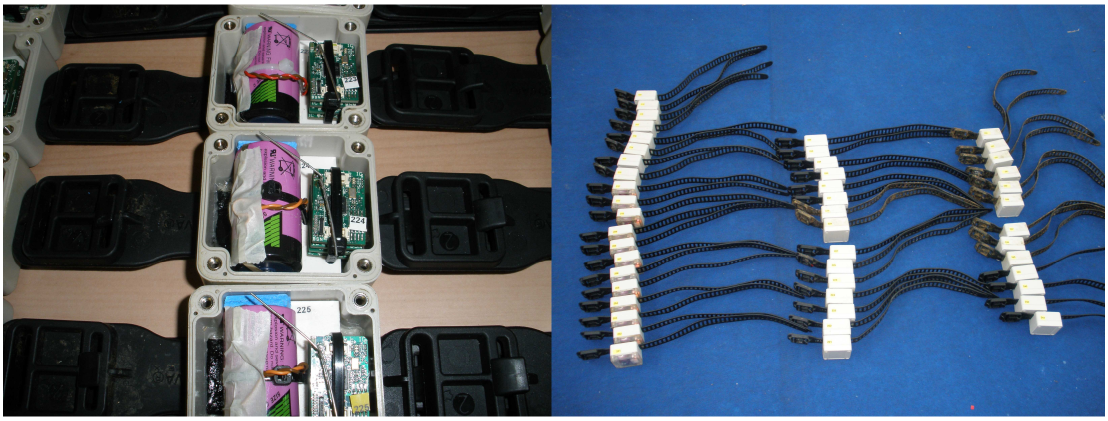


Figure 3.7 - Example of used ICL BSN nodes with Herd Control packaging

3.2.2 Ease of deployment (energy efficiency)

The firmware programming, configuration of the hardware and the attachment of a node to an animal is time consuming and labour-intensive. Removing a node for changing a battery, updating firmware or software or resetting a node is even more time consuming and labour-intensive. In practical applications it will be unacceptable to exchange a sensor more often than once a year.

During the first experiments there were a lot of problems because the software on the nodes hangs itself within a few hours after being activated. To create a workable application a reed contact was built in the sensor so the node could be reset without removing it from the animal.

The firmware of the nodes was updated several times finally resulting in robust firmware where the rebooting of a node was no longer needed.

A watchdog function was developed that checked what software modules were present on a node. If a software module was not / no longer present the module was automatically uploaded. This module was completed with a user interface giving an overview of the active modules on every node.

Insight in package loss (due to the animal laying on the node but also due to other causes) was improved by a user interface displaying the package loss per node per hour.

Due to an incorrect hardware/software configuration the energy use in the first experiments was extremely high. Batteries had to be exchanged every 2 or 3 days. After fixing the bugs and integration of the HC stack, and introduction of the algorithm on the node the batteries capacity was sufficient for supporting the node for a period of at least 10 days. The expectation is that that WiseMac and on node processing can contribute to a significant reduction of energy consumption.

3.2.3 Network stability and scalability

The network stability became an important topic in the iterative process of prototype development. Every time new components were introduced this issue came up. The debugging of the encountered problems took a lot of energy since it consist of a lot of different components that had to work together, a lot of different parties working on it and the non-visibility. During the project extra tools were developed to support the developers. Beside the ECA and uDSSP programming tools, also the SVN repository, the WASP Build System and the remote access became important. Also the simulation of specific circumstances helped to solve the issues when integrating the components. Encouraged by the successful large-scale herd control trial demonstrating the ability to monitor 30 (real and virtual) cows using in total 79 battery-powered nodes over 10 to 12 days, we performed a scaling exercise to assess whether it seems feasible to monitor a large number of cows over a period of a year without replacing the 8Ah batteries. Through this scaling exercise, we wanted to get a feeling on e.g. the:

- acceptable time between packets generated at application level
- acceptable number of battery-powered wearable sensor nodes
- required number and positioning of battery- and/or mains-powered forwarder nodes
- required number and positioning of sink nodes

that do comply within application Quality-of-Service constraints like:

- size (and energy capacity) of batteries
- packet delivery ratio
- packet delivery latency

Figure 3-8 illustrates two application classes and their associated time between (application) packets that are useful for a farm manager to monitor for both healthy cows (sending health reports to the farm information system at low periodicity) and cows that need attention (and send frequent health reports and information about their location).

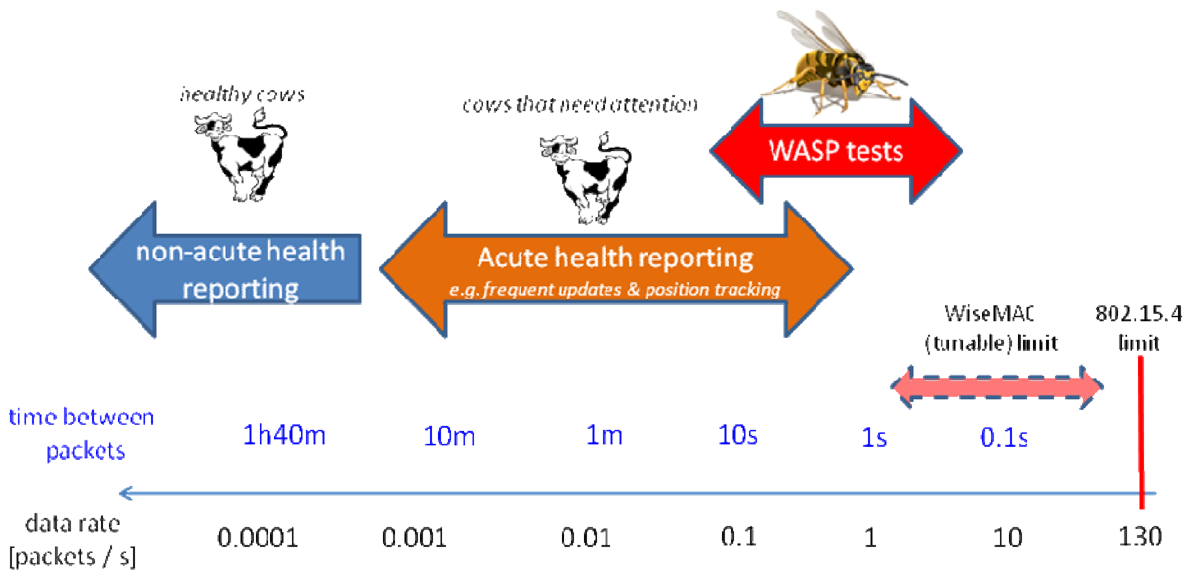


Figure 3.8 - Application ranges for herd control

The figure also indicates the application traffic typically generated in the WASP tests that are deliberately in a regime to stress test the sensor network capacity and the amount of data that can be send to the WASP gateway and enterprise system. The network capacity is fundamentally limited by the link & MAC layer and, in case of a (non-beacon) IEEE 802.15.4 MAC, typically is around 130 packets per second (assuming 120 Bytes per packet @ 250kbit/s and 50% overhead & efficiency). However, use of the (non-beacon) IEEE802.15.4 MAC, although fine for test bed development, is prohibited for long time operation as it requires the radio transceiver to be continuously active. This radio transceiver is the main power consumer (around 20mA) and, hence, main reasons why the 8Ah batteries in the HC test bed are emptied in 10 to 12 days. Other (beacon-based) IEEE820.15.4 or alternative MACs exist but these require additional time synchronization traffic especially in mobile environments. In parallel to test bed integration using the power-hungry (non-beacon) IEEE802.15.4 MAC, the low-power (non-beacon) WiseMAC protocol has been integrated in the WASP-OS (WOS) multi-threading environment. WiseMAC is a contention single-channel MAC protocol based on non-persistent carrier sense multiple accesses that is further described in section 5.6 of the WASP Development Handbook (D7.4). Depending on the WiseMAC sampling period, that determines the duty-cycling of radio transceiver between active and sleep mode, the throughput of WiseMAC can range typically between 1s (low but good for energy efficiency) and 50ms (higher capacity but at energy penalty) as illustrated in Figure 3.8.

We have performed various simulations to assess scalability or, actually, to explore the wide solutions space when optimizing application settings and deployment limitations. In short, we can rather easily monitor a large number of cows at proper QoS metrics (deliveray ratio >95%, latency <10s, and power < 3mW to enable battery-powered operation of one year) as long as the reporting time is a few times per day and network traffic (Gradient, FTSP, DV-Distance) is kept low e.g. by using the cross-layer optimization approaches described in deliverable D4.8. Further optimization of the Gradient / WiseMAC stack, to improve the current (very basic) duplicate filtering mechanism with mechanisms successfully integrated in the CCBR/TrawMAC stack, will improve QoS performance even further. When the deployment allows for mains-powered static forwarders, significant scaling and QoS benefits can be achieved by providing these forwarders with WiseMAC-High Availability which provide a much higher bandwidth to the sink). These results and conclusions are decribed in more detail in Section 4.1.2 of public deliverable D8.4 WASP final project summary.

3.2.4 Gateway integration

As outlined in the WASP public cookbook (D7.4) and in the deliverable D5.5, the WASP testbed environments can use so called hardware and software routers to forward the data coming from the sensor nodes to the IP domain using the WSN over IP protocol based on UDP unicast and multicast. The general concepts of this UDP tunnelling for WSN communication are briefly shown in figure 3.9 (see also D7.4):

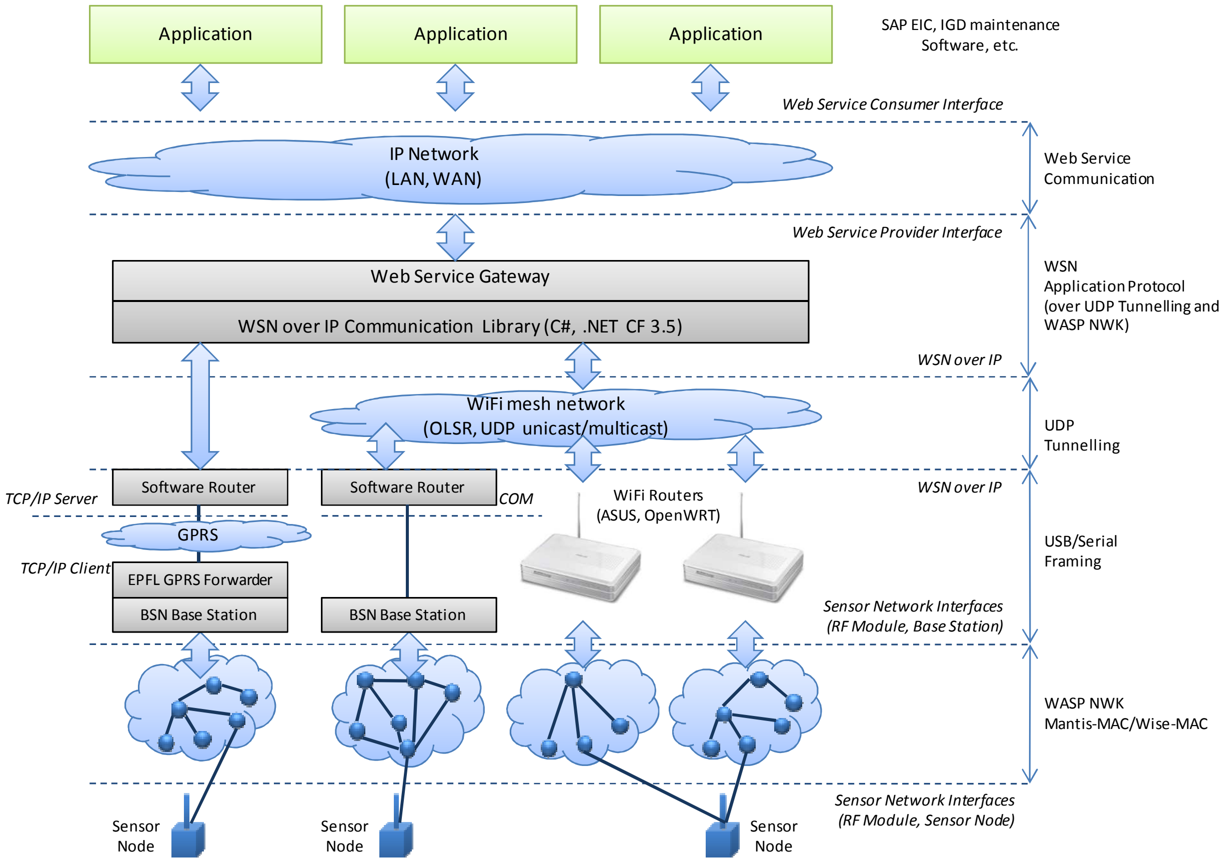


Figure 3.9 - General setup of the IP overlay network in WASP

It shows the different router types connected to WSNs on the one side and the IP infrastructure on the other side. This includes the EMIC Web Service Gateway (WSGW) and the end applications like the Enterprise Integration Component (EIC). The figure also shows the different options for connecting WSNs to the IP network. On the one hand, the Wi-Fi hardware routers (ASUS/OpenWRT) can be used. These routers also use their Wi-Fi interfaces to build a Wi-Fi mesh network via the OLSR protocol enabling a dynamically growing coverage area by just adding additional routers. The second way to bring a WSN into the IP domain is by using a software based router application on a standard PC. This software router can be also used as a TCP/IP server endpoint for the GPRS forwarder by EPFL.

The reasons for using both, hardware and software based routing solutions, have to do with the ongoing development within the WASP project. In a very early setup for the herd control testbed (HC), 802.15.4 and the hardware routers where used to be able to test the general setup in a real environment. This setup is illustrated in figure 3.10.

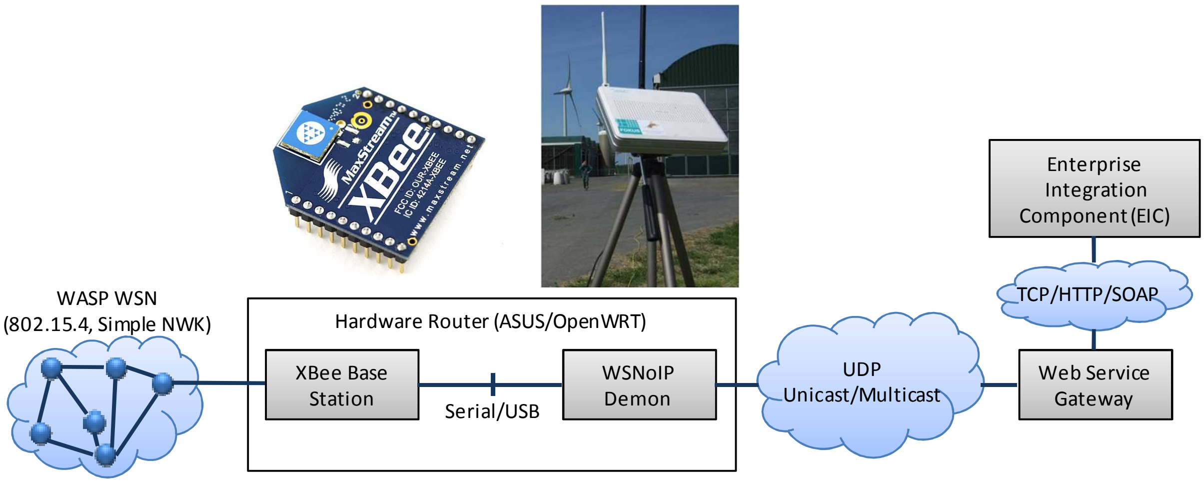


Figure 3.10 - Usage of hardware routers

The ASUS router uses an onboard XBee module as a base station to the 802.15.4 sensor network and has also a Wi-Fi interface for interconnecting different routers to increase the coverage of the network. In this setup only the Simple NWK API was supported over a single hop 802.15.4 MAC layer. The routers also had some capabilities to determine the signal strength (RSSI) and selected the closed router to a sensor node.

Later on, the WASP MAC protocols where available (WiseMAC, MANTIS) and it was possible to use a mesh based sensor network and also so called static nodes in the HC testbed that act as a forwarder. In this setup, it was not longer required to use several hardware routers. Thus, a software based solution has been selected that uses a BSN (ICL) HW with a dedicated base station software module connected to a PC or notebook. This setup is displayed in figure 3.11.

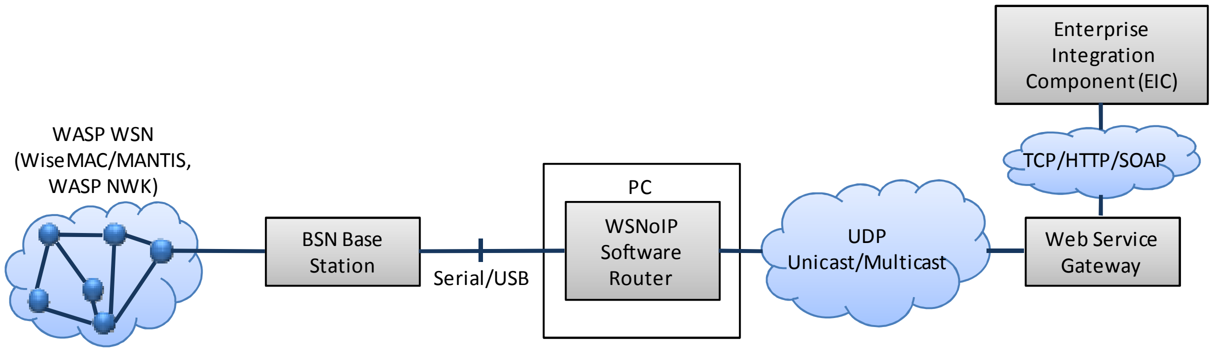


Figure 3.11 - Usage of software routers

Another benefit of the software router is that each partner can create a simple testing setup for the WSN over IP communication and also for the WSGW and EIB without having the hardware routers in the lab. Furthermore, the software router has a graphical user interface that allows the user to setup some parameters (e.g. serial port parameters or type of base station).

GPRS forwarder

In the herd control scenario, one of the goals was to enable remote monitoring and control of the herds when located in distant pastures. In such places, there is no infrastructure available (e.g., Internet connection, power supply) and it is impossible to use the Wi-Fi infrastructure set up in the barns due to the long distance.

The solution was to develop an interconnection device integrating both a BSN mote (to communicate with the sensor network) and a GPRS chip (to communicate with the servers), that we called a GPRS forwarder. The GPRS technology is widely available in almost all cellular networks around the world, and can be used to establish TCP/IP connections to the servers. The GPRS forwarder can then act as a repeater between the end points, allowing for seamless remote access to the sensor network.

The GPRS forwarder has been designed, produced, and tested during real-world experiments in Lelystad. It features the following characteristics:

- *Low-power consumption.* The embedded micro-controller is an MSP430, an ultra low-power MCU. It acts as the conductor for the forwarder, managing both TCP/IP connections and bi-directional communications between the attached BSN mote and the servers.
- *Duty-Cycled.* To save even more energy, TCP/IP connections can be duty-cycled. In this case, connections are regularly established for a few seconds. Once turned off, the GPRS chip does not consume any energy, thus effectively increasing the lifetime of the forwarder.
- *Small and light.* The GPRS forwarder fits in a small 18x12x6 cm plastic box, and is powered by 4 standard AA batteries that can be bought in about any store. It can thus be easily carried and installed wherever the herd is located.
- *Dynamic configuration*. The GPRS forwarder integrates an AT commands interpreter that can be used for dynamic reconfiguration. For instance, the duty-cycling mechanism may be temporarily disabled when remotely reprogramming the sensor motes.
- *Transparent*. The operation of the GPRS forwarder can be monitored from the server using AT commands. It is possible to query performance related values such as the number of packets dropped or the number of failed GPRS connection attempts.

With respect to the specific GPRS related issues the following lessons were learned.

- *Wired communications may be as unreliable as wireless ones.* The BSN mote had a buggy UART implementation, sometimes resulting in packet losses. An acknowledgment mechanism has been implemented to solve this issue.
- *Remote control is essential.* For some operations, the behaviour of the GPRS forwarder had to be dynamically modified. The AT commands interpreter was designed to fulfil this requirement.
- *Conditions of operation must be carefully considered.* The idea behind the GPRS forwarder was to operate the sensor network in remote places, exposed to the weather conditions. Particularly, rain and humidity are an issue with electronics. The enclosure of the GPRS forwarder was selected because it is IP67 compliant.
- *Keep it small and simple.* Debugging embedded systems has always been a non-trivial task. To avoid the biggest issues, the design of the software architecture must be kept as simple as possible.

3.2.5 Herd Control algorithm and location awareness

Data processing is needed to be able to detect health problems with focus on claw health and locomotion (the goal of scenario HC2). Acceleration data as measured by the sensors is not useful for the end-user (the farmer), it needs to be processed to information on behaviour and locomotion. Algorithms have been developed to convert the acceleration data into behaviour and locomotion information.

The behaviour can be monitored with accelerometers (used as a tilt sensor to measure the angle) in a wireless sensor network. The angle of a leg reflects the lying or standing behaviour, the angle of the head might reflect the eating behaviour. An experiment was carried during 50 days. Six cows were equipped with two 2D accelerometers, one attached to the neck and one attached to the right hind leg. The accelerometers were attached to wireless sensor nodes. The acceleration of the neck and leg was recorded every halve minute (average of seven measurements with 1 Hz measuring frequency). Based on calibration measurements, the acceleration of the leg and the neck were both transformed to the angle. A cow was standing when the angle of the leg was more than 45°, otherwise lying. The method to t ransform the acceleration to angle and behaviour appears to be appropriate [3.3]. A cow is walking when she is standing (according to the angle) and the variance of the seven variance measurements is above a threshold.

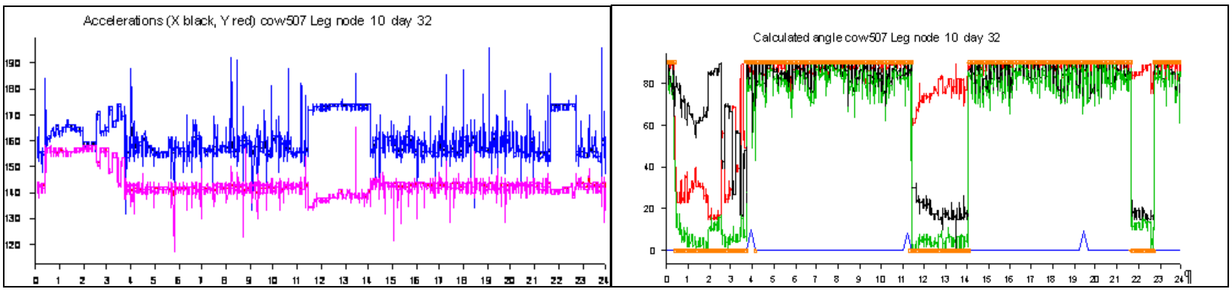


Figure 3.12 - Illustration of the conversion of measured acceleration (left) to angle and behaviour (right).

Accelerometers attached to a leg of the dairy cow can be used to record the locomotion. The feasibility of using 3D accelerometers in a wireless sensor network (WSN) for determining the locomotion of the feet of a cow was explored; a procedure to distinguish steps and to derive step parameters (length and time) was developed. It is possible to detect steps and to derive step parameters (but further research is needed to validate the results: [3.2].

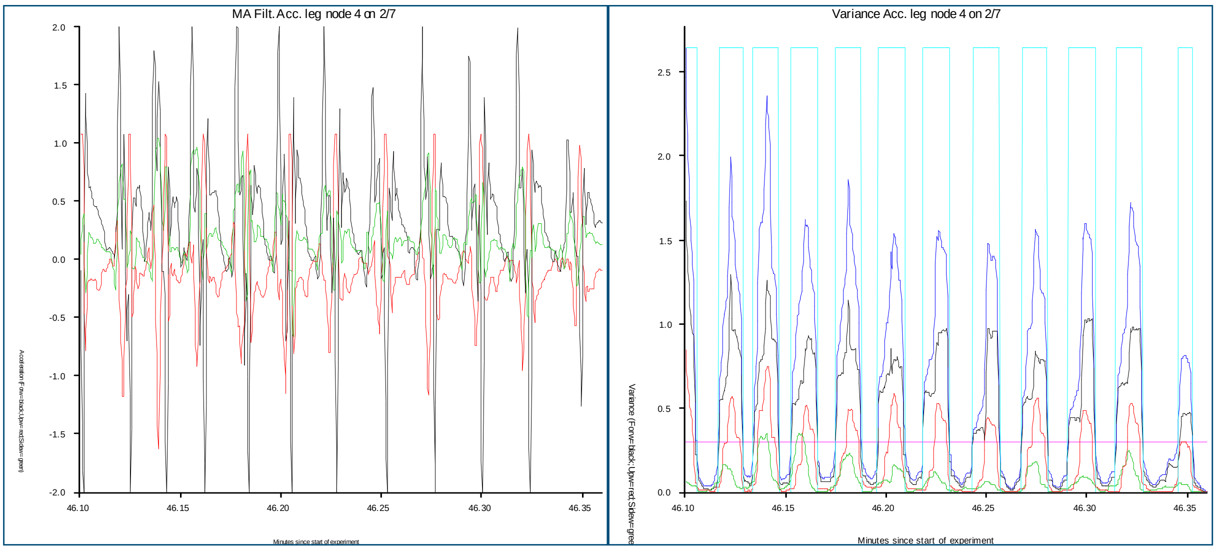


Figure 3.13 Illustration of the conversion of measured acceleration (left) to steps (right).

Validation

The data processing methods for determining the behaviour and locomotion have been validated by comparing the calculation results with result from video recording during one day (20 May 2010). Three cows were separated in a fenced-off part of the barn. Two nodes were attached to the hind legs of these cows (one left, one right) and acceleration data of these nodes were collected with approx. 50 Hz. Video observations were made with three camera's, two at a higher level (camera 1 and 2) and one at leg level (camera 3) (Figure 3.14).

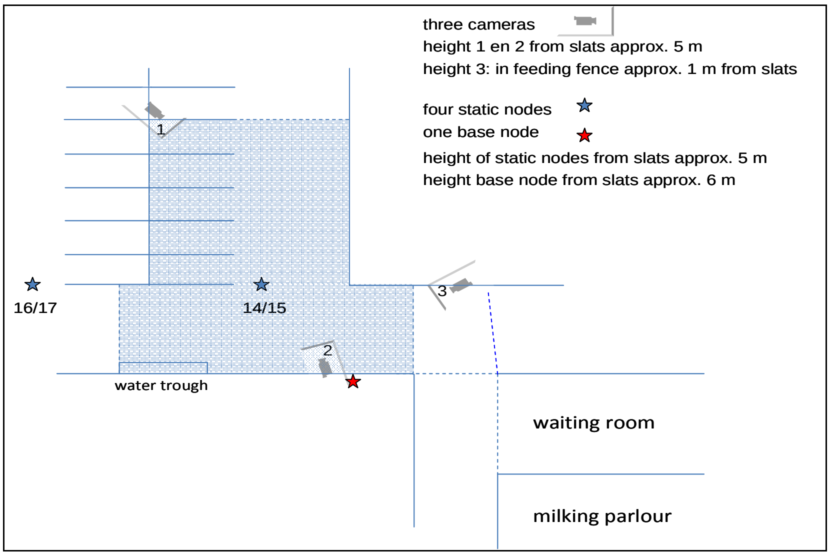


Figure 3.14 - Overview of the situation in the barn during the validation experiment.

During this validation experiment the acceleration data were processed off-line, all 50 Hz measurements were transmitted to the gateway and not processed on-node (as during normal functioning). Data processing was done off-line using the same algorithms as implemented normally on the node.

A comparison of the measured with the observed behaviour and locomotion showed a great similarity. An example is given in Figure 3.15.

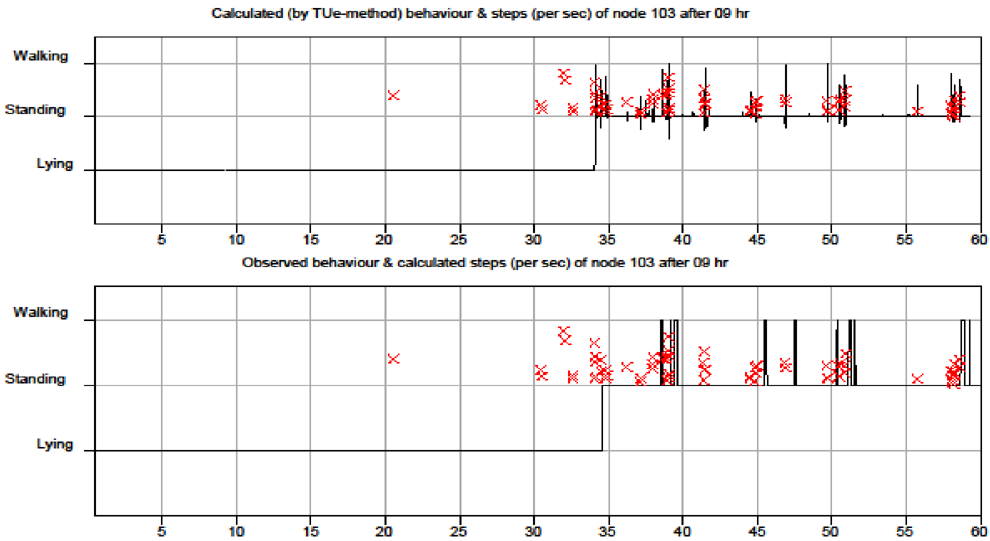


Figure 3.15 Illustration of the comparison of the calculated (upper) and observed (lower) behaviour for one node during one hour at May 20, 2010.

The validation data of 20 May 2010 have also been used to test the locomotion algorithms. Some step parameters per step can be calculated: contact time and move time, speed and displacement per step. An example is given in Figure 3.16. These parameters are based on the filtered values of the acceleration (median and moving average filter [3.2]). The acceleration, speed and displacement in Figure 3.16 are given in arbitrary values and have not yet been scaled properly.

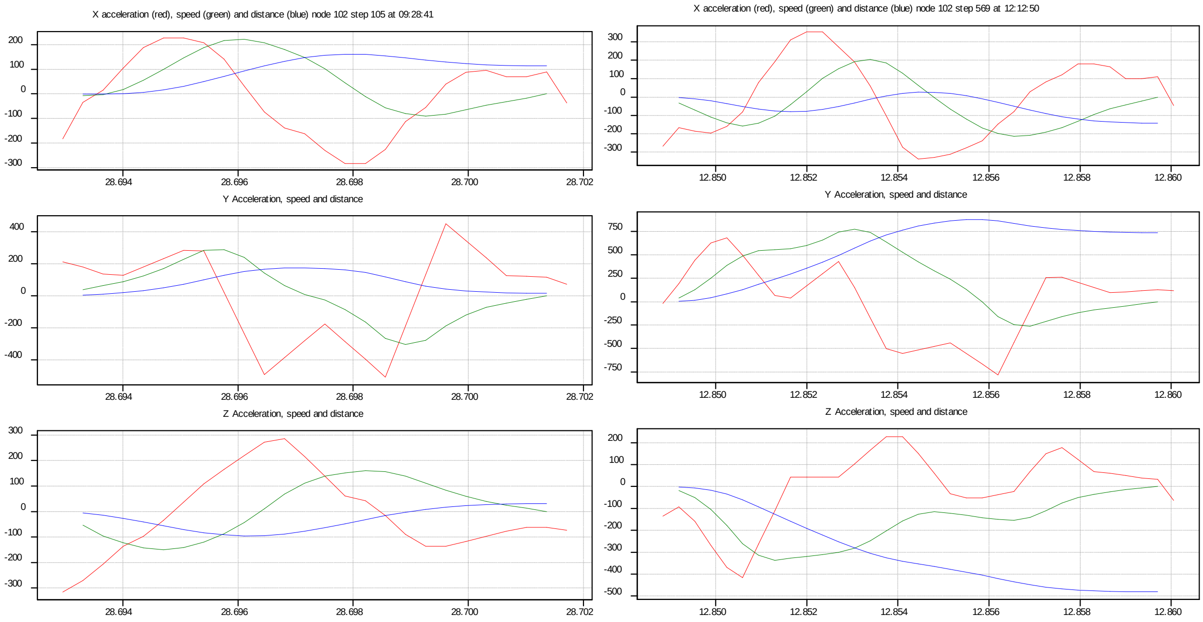


Figure 3.16 - Two examples of the calculated acceleration (red), speed (green) and displacement (blue) in three dimensions during two steps of Node 102 attached to Cow 3272.

3.2.6 EIC User interface

The user interface EIC is already described in chapter 2.3.5.1 were the focus is more on the Elderly Care situation. Although the EIC is setup for both situations in this chapter expectations for the more herd control specific user interface of the EIC is described. Almost all of them are incorporated in the final EIC version.

The current version of the EIC user interface is node-oriented: the user has to select a node, e.g. 'temperature on node node2'.

Wanted is a herd and cow-oriented approach where the user can successively (see Figure 3.17):

1) select a cow group;
2) select a cow within the selected group (only cows of that group are in the selection list);
3) select a node attached to the selected cow (only nodes of that cow are in the selection list).

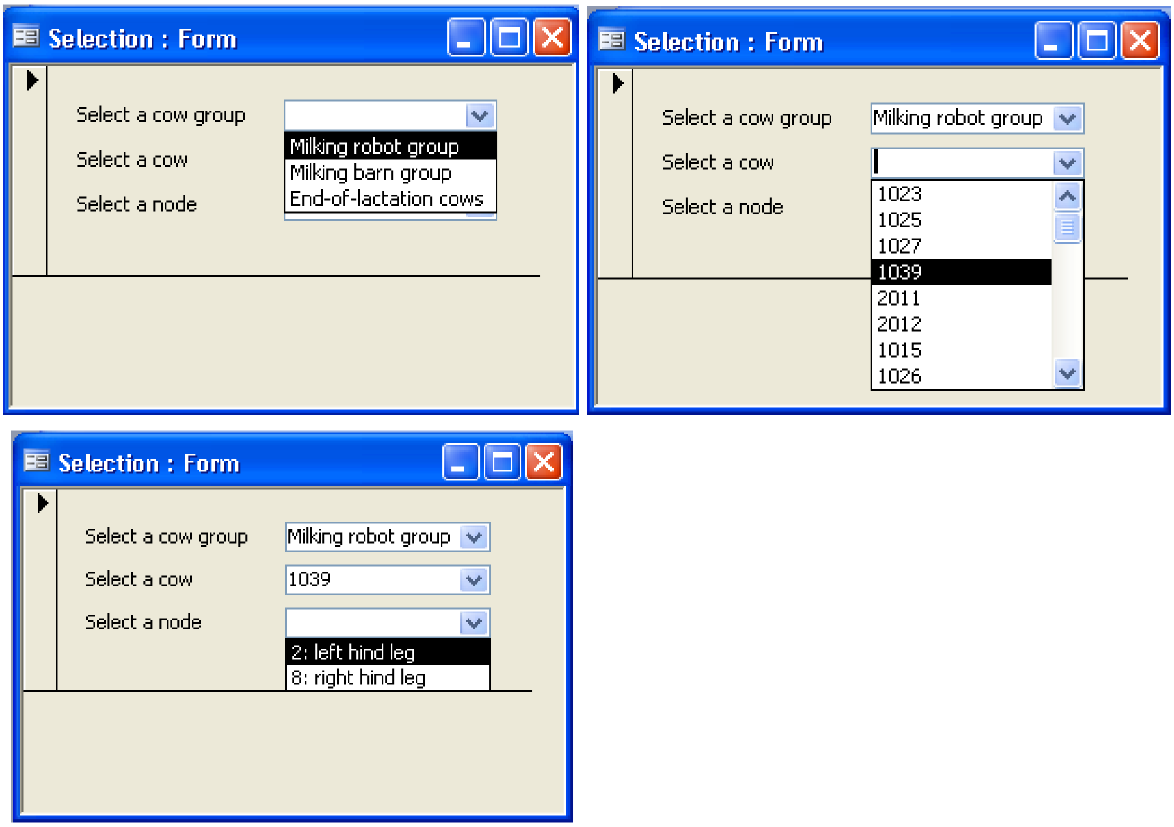


Figure 3.17 - Screen shots of the selection of cow group, cow within cow group and node within cow

A link between cows and groups is needed for this, as well as a link between nodes and cows. The link between cows and groups is static (for the duration of the experiment). The link between nodes and cows may vary; this problem can be handled in two ways:

1. add cow number to all messages from a node attached to a cow (cow number should be set when node is attached to cow) but this is different from the agreed data format and gives extra overhead in the messages;
2. relation between nodes and cow numbers fixed during the test bed and review but the relation has to be set before attaching the nodes or will only be known after the nodes have been attached to the cows.

The EIC could use a look-up table for the relation between groups, cows and nodes.

The data collected in the HC test bed depend on the mode chosen:

Mode 0: collect all measured data
Mode I: monitor lying behaviour
Mode II: monitor standing behaviour
Mode III: monitor walking behaviour
Mode IV: analyse locomotion during walking

Graphical presentation of data collected in Mode 0 is discussed in Paragraph 3.4, data collected in Mode I, II and II is discussed in Paragraph 3.3, data collected in Mode IV is discussed in Paragraph 3.2.

Step duration per node

Show graphs with plotted against time (see Figure 3.18):

- measurement value;
- lower limit defined as moving average minus two times the standard deviation;

- moving average of measured value (calculated over a certain time lag);
- upper limit defined as moving average plus two times the standard deviation.

Measured values below the lower limit or above the upper limit (outside confidence interval) should be alerted, giving them a different colour (as in Figure 3.18) is not possible, but an additional graph displaying whether or not there is an alert may an option.

The shown time interval should be the last 24 hours with the possibility to show other time intervals, up to the last 14 days. The time lag for calculating the moving average, e.g. the last 10 measurements, can be adapted.

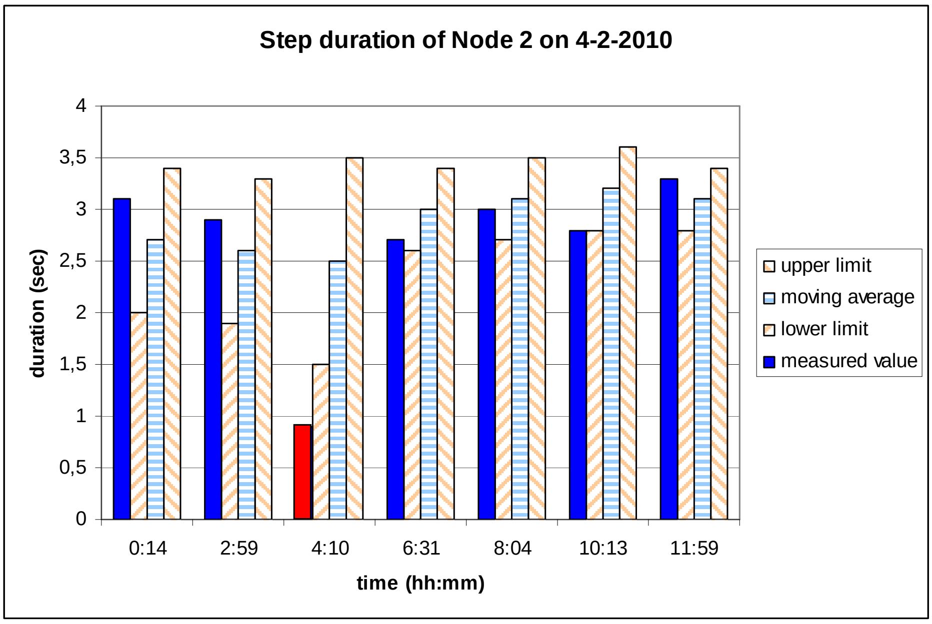


Figure 3.18 - Graph of step duration per node: measured value, lower limit, moving average, upper limit against time

Step duration per cow

Show graphs with plotted against time (see Figure 3.19):

- left/right ratio of step duration;
- moving average of left/right ratio of step duration.

The left/right ratio is defined as the latest value of the step duration of the left leg divided by the latest value of the step duration of the right leg. The values of the left and right leg will be on different moments in general, for the ratio the latest values should be used and the time difference should not be too big.

Measured values below the lower limit 0.9 or above the upper limit 1.1 should be alerted, e.g. in an additional alert graph. The thresholds will be adaptable. It is not possible to show the threshold lines in the graph. The shown time interval should be the last 24 hours with the possibility to show other time intervals, up to the last 14 days.

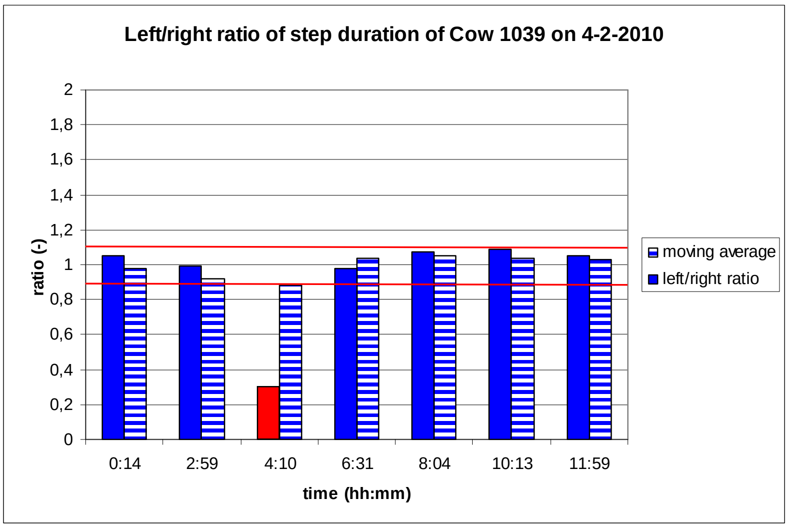


Figure 3.19- Graph of step duration per cow: left/right ratio and moving average of left/right ratio

Behaviour per cow

Show graphs with plotted against date (see Figure 3.20):

- percentage of time lying/standing per day (based on the length of the standing and lying bouts);
- number of lying bouts per day;
- length of standing bouts per day;
- length of lying bouts per day.

A bout is a period in which a cow continuously is occupied with the same activity (e.g. lying)

An alert should be given when:

- percentage of time lying less than 40% or more than 60%
- number of lying bouts more than 20
- average length of lying bouts less than 30 minutes
- length of a lying bout more than 2 hours
- average length of standing bouts less than 40 minutes
- length of a standing bout more than 3 hours

The alerts may be presented in additional graphs (and not by different colours as in Figure 3.20).

The thresholds for the alerts will be adaptable.

The shown time interval should be the last week with the possibility to show other time intervals, up to the last month.

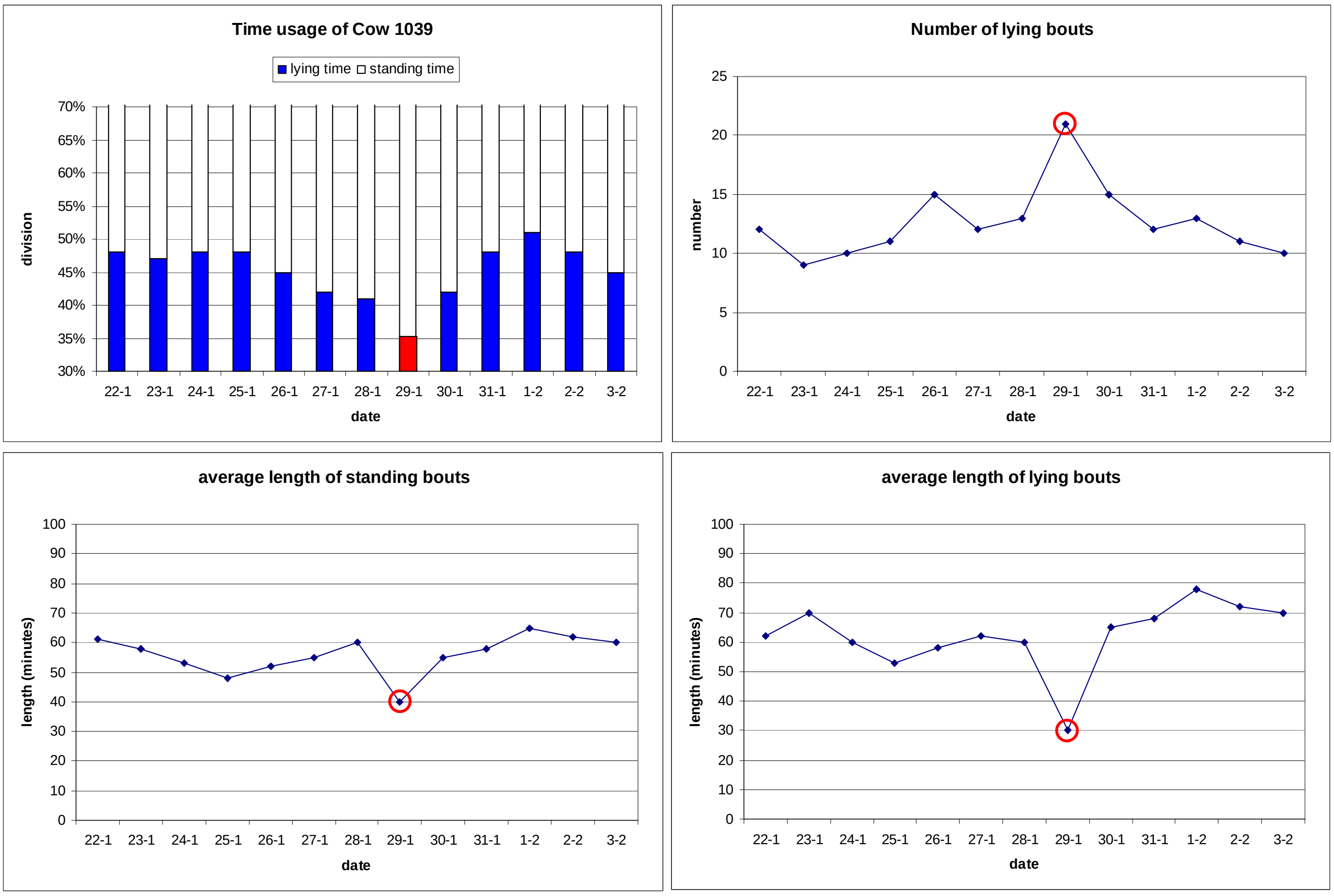


Figure 3.20 - Graph of behaviour per cow: time usage, number of lying bouts, length of standing bouts and length of lying bouts

Behaviour per herd

Show graphs with plotted for all cows in cow group (see Figure 3.21):

- percentage of time lying/standing
- number of lying bouts
- length of standing bouts
- length of lying bouts

An alert should be given when:

- percentage of time lying different from group average (difference more 2 times standard deviation);
- number of lying bouts different from group average (difference more 2 times standard deviation);
- length of lying bouts different from group average (difference more 2 times standard deviation);
- length of standing bouts different from group average (difference more 2 times standard deviation).

Alerts may be presented in additional graphs (and not with different colours as in Figure 3.21).

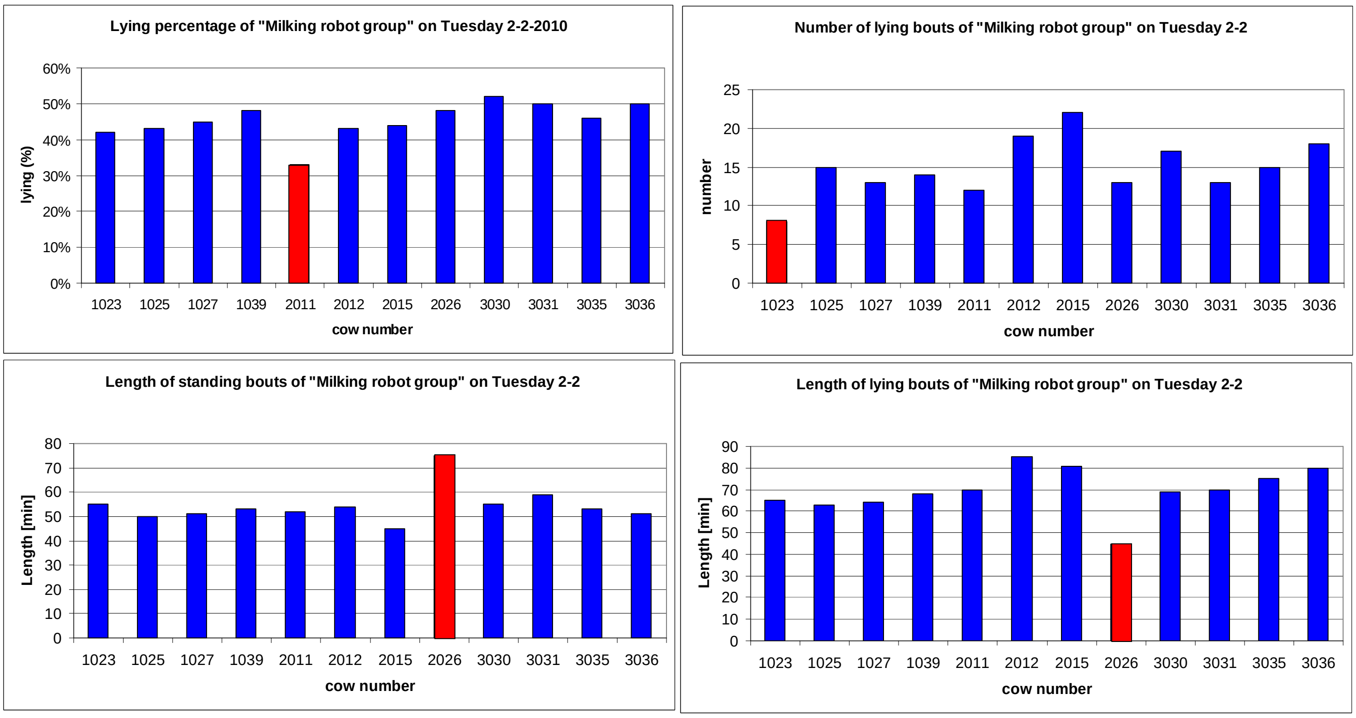


Figure 3.21 - Graph of behaviour per cow group: time usage, number of lying bouts, length of standing bouts and length of lying bouts

Collected data

In Mode 0 all measured data is collected. The graphical presentation can be done by plotting the data against time, e.g. as in the current version of the EIC for the Elderly Care test bed (see Figure 3.22).

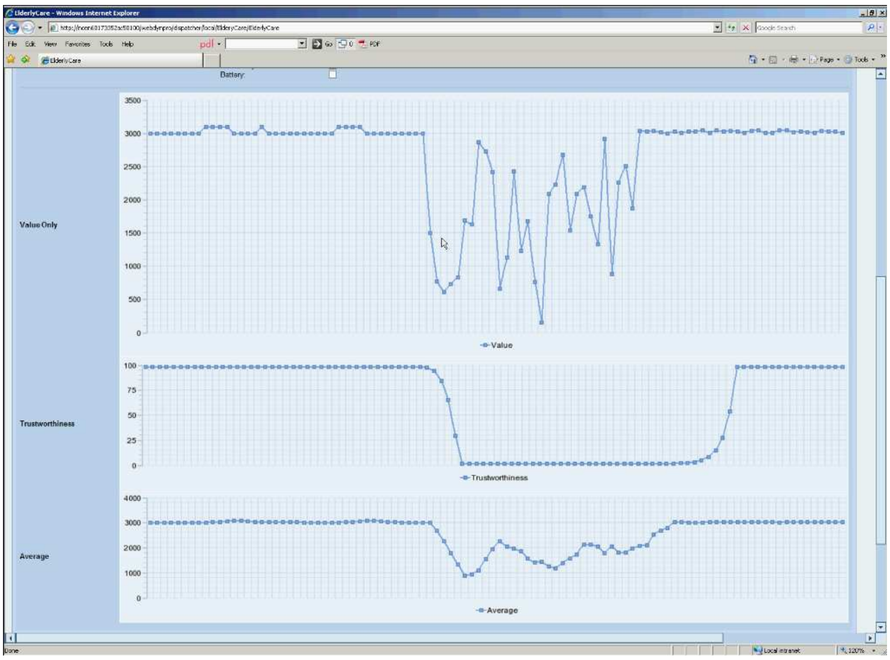


Figure 3.22 Screen dump of the EIC for the Elderly Care test bed.

3.2.7 Prototyping and testing

Although originally not planned from July 2008 onwards, biannual integration meetings have been organized for both EC and HC test beds, to gradually introduce new components and move from technology testing towards actual scenario testing. Four integration meetings per year proved sufficient to make good progress although, with the core integrators also involved in WP-specific tasks, did provide some timing conflicts. The preparation and the actual organization of the integration meetings were driven by the test bed coordinators and project manager and, although each meeting had clear targets, the meeting schedules needed to be flexible to cope with unsuspected issues as those mentioned in the sections above. These issues basically reflected a natural process to mature technologies and align the development environments used by individual partners.

Nevertheless, meetings could have been more efficient if:

- A single person would have been appointed that has the technical skills to run a well-defined basic integration test prior to an upcoming test bed meeting to identify critical bugs in advance. In WASP, we relied on volunteers to do so next to their other test bed commitments which sometimes provided timing conflicts. We tried this concept in task 6.9 ‘testing components and requirements’, but had to decide not to continue this.
- We really experience the gap between the technological developers and the scenario driven end users. Too few people could combine both worlds and drive the development form a holistic concepts. Nevertheless big steps were made and the end users (researchers) and the developers worked quite well together.
- The feeling is that prototyping and test beds revealed the real issues to be solved for large scale sensor networks and that time was ‘lost’ in finding the theoretical solutions. As an example for long the option were kept open for Mantis MAC or FreeRTOS as operating system to be used in the test bed. At the moment the decision was made for Mantis MAC everybody agreed and further work was aligned according to that decision.

For a large and complex IP project like WASP with challenging (test bed) ambitions, it is strongly advisable to appoint a dedicated technical director to drive and keep momentum in the alignment and integration efforts, especially across work packages. In WASP, technical alignment in general and test bed alignment in particular involved quite a few topics and partners including the architectural team (to prepare alignment proposals), the test bed integration teams (to assess and implement technical alignments), the WP leaders (adjusting WP commitments to integration needs), the test bed coordinators (evaluating achievements and updating plans for next meeting) and the project manager (to keep track of agreed activities and open issues). This shared approach worked out in the end but integration would have been more efficient when a dedicated technical director, able to fully focus on the technical details of all parts of the WASP system, would have been appointed right from the start of the project to keep momentum and maintain full overview of all issues.

3.3 Herd Control discussion

In this paragraph the overall lessons learned form the Herd Control prototyping and testing are given form the perspective of an end user. The basic WASP principles are the same for the elderly care prototype and for the herd control prototype. Using the programming model and working with the service ecosystem are coherent. In building the prototype and testing on system level different accents were put on the prototypes. The herd control prototype focuses on ‘large’ networks, and on testing the quick data transfer by using different strategies: static sensor nodes, router nodes and GPRS nodes. The following challenges were therefore incorporated in the herd control scenario.

Challenge: Programming of the nodes:

The application-specific services needed for using a node in the Herd Control scenario can easily be selected from the available services. In principle four services are visible for the technology user:

- Information about the locomotion of the cows
- Overview of what the cows are doing (walking, standing, lying).
- Overview of the daily behaviour pattern of lying, standing and walking and, if possible, eating, drinking ruminating and resting.
- Information about were cows are located (rough indication inside and outside barn).

Most of the Herd Control measurements are related to a moving cow. If a cow is not walking there is no necessity of measuring the activity, locomotion or location of the animal.

→ *The programming environment that was based on ECA and uDSSP worked well for the HC scenario. The whole scenario was integrated in one programme. The challenge to work with an ecosystem of services succeeded. The programming of the prototype was mainly done by the developers from TUE, EMIC and SAP, but end users were able to maintain and operate the system in the test bed. This includes flashing, installing and changing of nodes and network infrastructure, and starting all the needed programs. With regard to the functionality the system so far is restricted to inside barn following of the cows. Differentiation between walking, standing and lying (different node modes) is possible and also it was possible to setup some different network situations (different network modes). Eating, drinking, ruminating and resting were not incorporated in the prototype. There was enough challenge in the prototype for the development of the WASP concept.*

Challenge: State-based triggering:

At regular intervals is checked if the behaviour of the cow has changed. The duration of the interval is based upon the behaviour that was recorded during the previous observation. The duration of the interval can be changed based on the cow's state: e.g. 1 minute when lying, 5 seconds when standing and 1 second when walking. If precise location measurement is activated then when walking the new location of the animal shall be recorded at regular intervals (e.g. 5 seconds). Twice every day the locomotion pattern should be recorded when the animal is walking. The criteria for starting (state based triggering) a locomotion pattern measurement could be: the animal is walking and the last correct recording of the locomotion pattern is more than 10 hours ago or is initiated by human interaction when the user wants to know the current locomotion characteristics.

→ *During the development of the prototype these elements were worked out and tested successfully.*

Challenge: Online re-programming:

When attaching the sensor nodes to an animal, the information of the sensor node must be linked to the animal-id and to the part of the body of the animal were the node is attached to (e.g. head or right hind leg). This information shall be linked by using the radio link.

Software updates shall be made effective by using publish-subscribe. The updates shall be made effective without interaction between farmer and cow.

→ *The first part of the online re-programming is not implemented fully. In the EIC a connection is made between the node-id and the cow-id. The connection through the radio link is not implemented. However, the second part is successfully implemented. Remote parameter setting and sending updates is possible and has been shown several times. Also preparation and flashing of nodes has become 'routine' work.*

Challenge: On node processing (data compressing):

The nodes hardware shall be capable of interfacing 3-D accelerometers with a frequency of 50 Hz. The node software operates stand alone and is capable to process the sensor data. The parameters characterizing the measured data are transmitted by the radio link.

→ *This has been developed and tested successfully. The algorithm is transforming the 3D sensor information into activity and locomotion information. This algorithm can be used for post-processing in the EIC and for on-node processing. The number of data packages to send is reduced with a factor 150. Difference between continuous measurements with 50 Hz compared to sending processed locomotion data per 'minute', depending on node mode.*

Challenge: Communication tools:

Cows can be housed inside our outside. Inside the barn static sensor nodes can be used for interfacing the nodes. Outside router nodes (Wi-Fi for middle range) or GPRS nodes (long range) gateway can be used for interfacing the WSN-nodes.

→ *Although we did not used the system outside the barn we implemented and tested different router implementations. The final prototype uses the SW router and the GPRS router. The GPRS demo already in 2009 resulted in a working and stable communication. Improvements were made in the last period on data handling protocols when the GPRS is not continuous on line. In the final prototype the SW routers are used and they function quite well.*

Challenge: Location awareness:

Based upon the received signal strength or the routing (based upon the configuration of the network of the static sensor nodes, router nodes and GPRS nodes) of the data the position of the animal will be estimated.

→ *This functionality was foreseen to be part of the chosen HC-stack solution. DV-distance should be implemented in the prototype. In the final demo prototype DV-distance has not been implemented in the HC stack, and it is only shown in a side demo together with a maintenance service.*

Challenge: Scaling effects:

The size of a herd can be far beyond 100 animals. If every animal receives one or two nodes this will result in a density of several hundreds of nodes. This requires efficient use of the WSN network.

→ *This was the biggest challenge for the HC prototype. The choices made in the project were also based on the challenge to be able to monitor moving cows is a difficult area. The effort to build a WSN network and a suitable programme was lowered in every iteration step. Flexibility in programming and remote control to the network makes deployment of large scale networks easier. With concern to the energy consumption of the nodes improvements were made. By using the algorithm on the node itself the data transfer, and therefore also the energy use, was reduced. With respect to energy use also much improvement is expected form the HC stack. Gradient routing on WiseMac should make it possible to go up to a couple of months. However we did not succeed to create a stable large scale network with these settings in time. The alternative is working with gradient routing on MantisMac. With these settings we created a stable network that we could scale up to more than 70 nodes. The disadvantage is that the longevity of the batteries was limited to 10 days. So it was possible to setup a large scale network with the* *appropriate programs to observe moving cows and determine their behaviour in time, but still improvements are needed on the energy use of the system.*

4 Simulations of automotive scenarios

According to project planning, real (test bed) implementations are being deployed for the herd control and elderly care application domains. For the (third) automotive domain, the applicability of WASP solutions is tested and assessed by performing simulations for some selected scenarios. The following sections of this chapter report the description of the simulations performed and the results collected.

An extended list of potential scenarios has been described earlier in D6.2 [4.2] and, from this list, we have selected in-vehicle scenarios which are recognized, at present, as the most promising on the short to medium term. We consider a network composed from several wireless sensor nodes enabling functionalities nowadays available through wired sensor solutions like light and temperature monitoring, passenger presence detection, side window and rear-view mirror control. Goal is to obtain a detailed insight of in-vehicle wireless sensor network performance (e.g. end-to-end latency), reliability (delivery ratio), power consumption (node lifetime) and overall node activity for related use cases in order to find out whether (and which) wired sensors may be replaced by wireless sensors.

Another automotive scenario that we consider for assessment through simulations concerns "Wireless sensors and data replication within a tunnel" described in D6.2 as well.

The BAN stack is adopted for simulations of the in-vehicles network applications whereas the HC stack is used for the tunnel network. Different simulation models are defined and tested for all the selected use cases in order to evaluate network performance and identify weak points. The results coming from the simulations will be compared with the use cases requirements.

The analysis is not focussed on very specific sensor applications. The design of the networks considers a generic deployment of nodes aimed to assess the sustainability of the overall node activity rather than to tune minutely the configuration parameters of a particular sensor node. The sensors referred in the trial have been selected just to set some domain-specific requirements. The number of nodes in the network, their deployment, the transmission rates and the interaction rules, have been set as generic as possible in order to take into account the majority of possible cases and sensors.

Reference applications, simulation model and environment, overall network design, tests performed and results are here fully described.

4.1 Scenarios description

4.1.1 Wireless sensors and data replication within a tunnel

This paragraph provides a brief description of the use cases that have been selected to experiment the WASP protocol stack architecture for the monitoring inside a long tunnel.

The tunnel is already equipped with an existing incident detection infrastructure, but the tunnel management decided to add a second set of wireless sensors (smoke and temperature), in order to increase the number of the monitored points.

Figure 4.1 - Sensor monitoring inside a road tunnel

Wireless sensors must be able to interact with the existing infrastructure through a number of gateways, provided with both wired and wireless interfaces.

4.1.1.1 Structural and environmental monitoring

From this cluster of applications, the following are taken into account:

UC1 - the issue of an alarm from a sensor in a normal situation (i.e., the network is working properly). Whenever a sensor node detects an abnormal condition, it must issue an alarm.

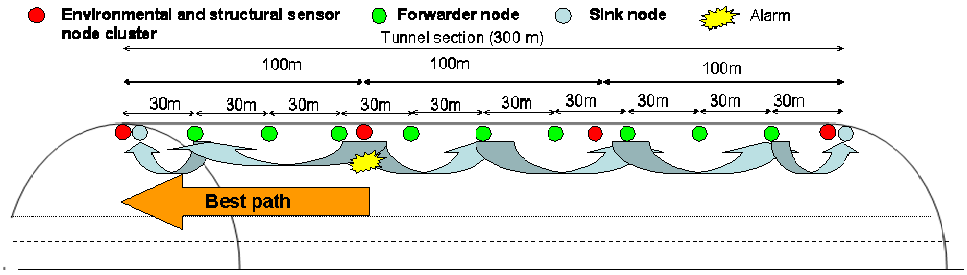


Figure 4.2 - Alarm from a sensor in a normal situation

The node must perform all the actions needed to ensure the reception of the alarm by the tunnel management centre, until the centre itself does not cancel the alarm condition.

UC2 - the issue of an alarm from a sensor when one or more forwarders are compromised. If the closest forwarder, used in normal conditions to reach the closest sink is not operational, the WSN must take actions in order to communicate with the tunnel management centre using an alternative path.

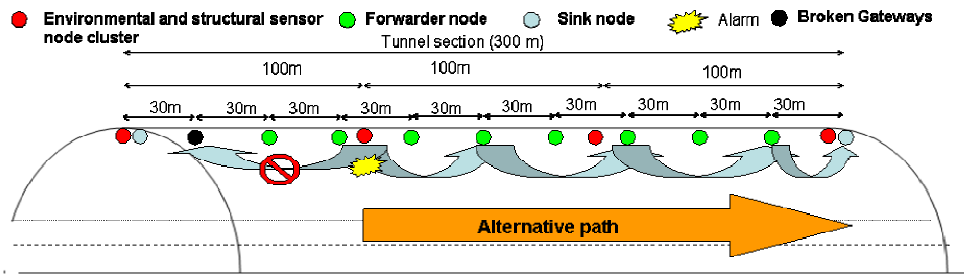


Figure 4.3 - Alarm from a sensor when one or more forwarders are compromised

UC3 - the periodic check of nodes status and battery level performed by the network itself. The network should be able to periodically check the status of all nodes and sensors in an autonomous fashion.

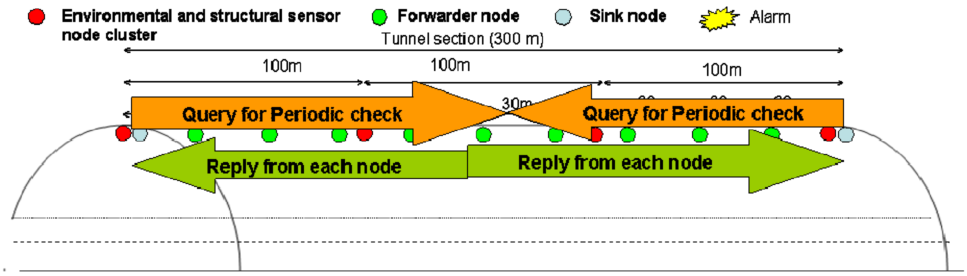


Figure 4.4 - Periodic check of nodes status and battery level

The periodic check should be designed in order to avoid false alarms and to allow an easy process for node recovery in case of failure.

4.1.1.2 In-situ interactions

In order to make the scenario more complete, the interactions with mobile vehicles and maintenance operators are taken into account.

First of all, maintenance operators must be able to access the data of every node in order to check the status and the correct functioning of sensor devices.

Secondly, emergency vehicles (low-mobility) must be able to work as gateway in emergency situation and collect all the information available in the WSN, without interfering with the data exchange between the WSN and the central infrastructure.

In order to consider these features, one additional use case is here investigated:

UC4 - check of the sensors status and battery level by maintenance operators and service/emergency vehicles.

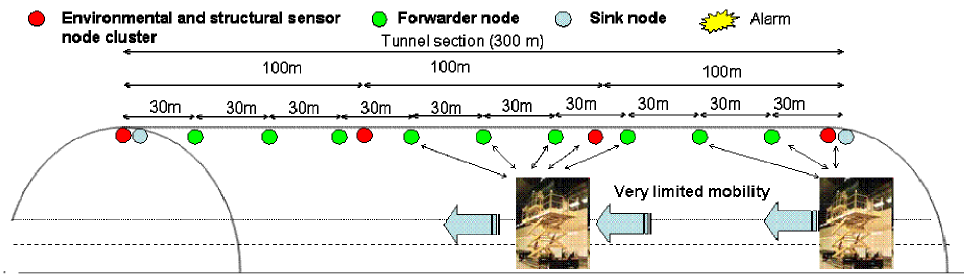


Figure 4.5 - Check of sensors status and battery level sensors by the maintenance operators.

Assistance vehicles should be able to perform a direct check of nodes status using a direct connection. This procedure could be used during the network deployment and for maintenance purposes. The collection should be possible even if some part of the network is not operational, because of the emergency situation. This function should not interfere with the connection between the sensor nodes and the control centre.

4.1.1.3 Vehicle info

Finally, the interaction between the private vehicles and the wireless network infrastructure is tested.

UC5 - Vehicles send multicast messages with info of speed, engine status and/or (flat) tire pressure.

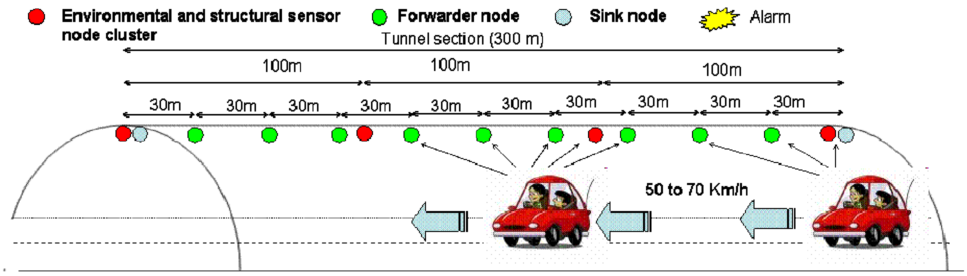


Figure 4.6 - Interaction between vehicles and infrastructure

4.1.2 In-vehicle sensor networks

Applications are grouped in three different clusters:

- Second generation of vehicle E/E architecture. This cluster deals with the introduction of wireless technology into the vehicle architecture in order to realize functions already available/possible but accordingly with a different approach.

Figure 4.7 - In-vehicle sensor networks

  Amongst them:

    - the passenger presence sensor,
    - the steering wheel buttons,
    - internal lights,
    - side rear view mirror
    - dashboard buttons;
- Introduction of new functions. Wireless is seen as the enabling technology for the functionalities nowadays not allowed mainly due to cabling, such as:
    - brake monitoring system,
    - intelligent tyre;
- Personalization cluster, aimed to group functions which are related to the capability of vehicles to follow user desire and personal profile, like for example:
    - User identification (when gets in car),
    - User profile management,
    - interaction with biomedical sensor (V2D),
    - HMI personalization (user can plug button in the most comfortable position)
    - HMI extension (new features introduced as aftermarket needs HMI)

4.1.2.1 Intra-vehicle network

A network composed of several wireless sensor nodes amongst those listed, is here considered. The idea is to combine clusters 1 and 3, in order to realize a scenario where different nodes (sensors and actuators) can cooperate to enhance the capability of vehicles to follow user desire and personal profile. The network includes:

- passenger presence detection sensor;
- internal light monitoring sensor (compartment);

- internal temperature monitoring sensor (compartment);
- rear view mirrors actuator,
- biomedical sensors (V2D);

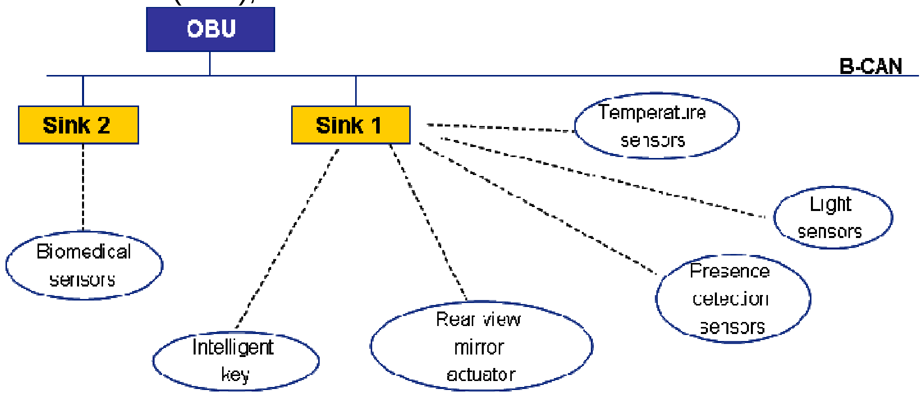


Figure 4.8 - Intra-vehicle network architecture

Challenging architecture:

- different types of nodes: from sensors to actuators with different constraints in terms of data rate, latency, power consumption
- variable number of nodes (scalability, network reconfiguration).

4.1.2.2 Tire Pressure Monitoring System

Sensors are inserted inside the tires and collect information about pressure and temperature (air and internal tire surface). The information is periodically sent to the vehicle, but an internal logic is able to trigger an alarm in presence of a failure.

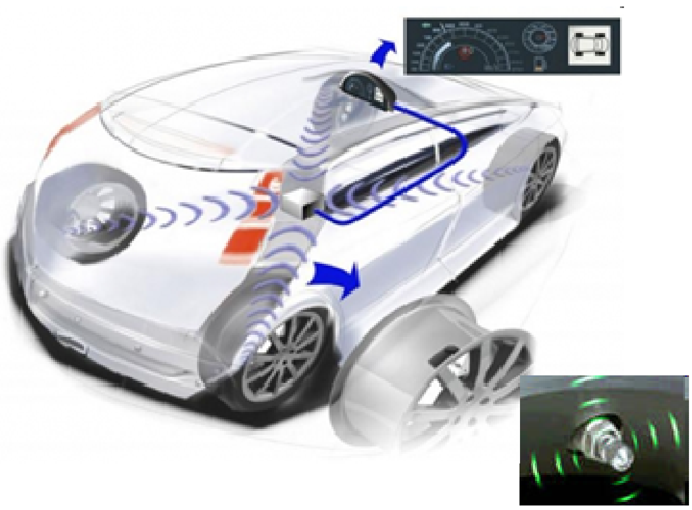


Figure 4.9 - Tire Pressure Monitoring System

The device must be able to work autonomously at least for the lifecycle of the tire (3-7 years).

TPMS consists of the following components:

- One Tire Pressure Module (TPM)
- Four sensors (one per tire)
- One Instruments Panel Cluster (IPC)

The system monitors the tires pressure and gives to the user relevant information about variations of the tire pressure through the Instrument Panel Cluster (IPC).

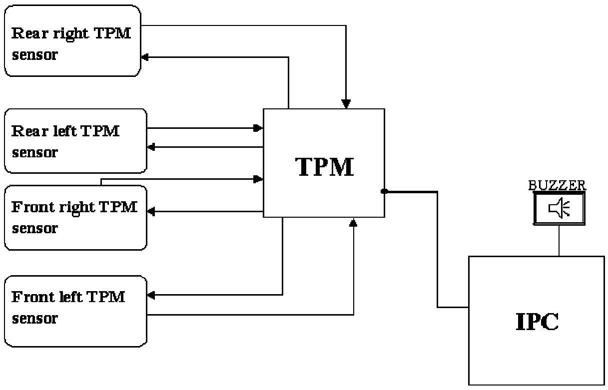


Figure 4.10 - TPMS architecture

TPM is an ECU able to receive via RF data about pressure/temperature transmitted from the sensors inside the wheels. In order to wake up the sensors (normally in sleep mode) the ECU has to send them an initiator message.

Definitively, TPM acts as sink node performing the following functions:

- Wake up each sensor through an activation message.
- Receiving information from sensors about pressure and temperature.

The wheel sensors are usually in "sleep mode". In this working condition the sensors measure the tire pressure and temperature but they do not transmit any information to save the internal battery. The TPM wakes up the sensor which then changes from Sleep mode to Normal mode working condition and transmits its ID, tire pressure and temperature each 60 seconds.
In alert mode, tyre pressure and temperature data shall be transmitted every 4 seconds for 60 seconds duration. Two use cases have been simulated: the Normal Mode functioning and the Alert Mode.

4.2 Models and tools

The scope of this Chapter is the definition of the guidelines to perform the simulation for the protocol stacks assessment in the automotive applications. This evaluation involves the implementation of all the protocols stacks developed in the project, optimized toward the elderly care and herd control applications but here tested with reference to automotive use cases. As specified in the introduction of chapter 4, the BAN stack is adopted for simulations of the in-vehicles network applications whereas the HC stack is used for the tunnel network.

4.2.1 Evaluation metrics

In performing our simulations, the following metrics are taken into account (see [4.1] for more details):

- **Power consumption:** WSN nodes are usually powered by batteries. Power consumption during node operation determines battery life. Power consumption depends on the different hardware and software components in a WSN node and their various activities. In order to determine the life of the battery, we must measure the power consumption of a node that is active in a network. That is, we must know the power consumption and time duration for node activities including computations, and RF transmission and reception;

- ***Data packet delivery ratio***: the number of data packets received by the destination nodes in a certain period of time, divided by the number of packet transmitted by the source nodes in that period. Here, we look at the percentage of packets sent by the sensor nodes that are properly received by the two sinks. The easiest way to calculate this percentage is the following:
  - Each sensor node, which is characterized by a proper ID (nodeID) sends with the predefined transmission rate, a message with an incremental ID (MessageID).
  - In each sink node, a variable increases of one unit every time that a message from the sensor nodes is properly received. At the end of the trial, the Delivery ratio is calculated simply dividing the final value of the counting variable for the number of packet sent.
- ***End-to-end latency:*** how much time it takes for a packet of data to get from a sensor to a sink.

All these metrics are the result of the amount of different contributions. Some of these contributions depend on hardware characteristics. Others depend on software and the configuration of the device. The aim of our tests is to draw the average values of power consumption, end-to-end latency and delivery ratio for every protocol stack and then to compare the results.

4.2.2 Tools

OMNet++ 4 [4.3] with Mobility Framework 2.0p3 [4.4] have been selected as a common platform for simulation of the BAN and the HC stacks.

OMNet++ 4 is the latest release of the OMNet++ discrete event simulator. OMNet++ is an extensible, modular, component-based C++ simulation library and framework, which is rapidly becoming a preferred simulation platform in the scientific community. Mobility Framework is a library specific for simulations of wireless sensor networks and its construction is partly financed by the WASP project.

This simulation framework is chosen as it is extremely easy to use and customize, and provides an integrated set of tools of design, development and analysis which actually shorten the time to perform the simulation phase.

Within this simulation framework of OMNet++ 4 and Mobility Framework, models of actual hardware and software used in the WSN can be developed and analysed by simulations.

4.2.3 Models

In this section the models created for the BAN and the HC stack are described.

OMNet++4 and Mobility Framework are currently released including most of the needed models to simulate the test scenarios described in Chapter 4.1.

The simulated BAN stack is based on IEEE 802.15.4 MAC and Pairwise routing protocol which is a modified version of NSafeLinks. Pairwise and NSafeLinks are both thought for static networks, support tree routing from sensors to a single sink, and create routing tables counting only on neighbourhood information. Pairwise extends NSafeLinks in terms of robustness against link failures and support to reverse tree routing from the sink to sensors; NSafeLinks uses flooding in the reverse direction instead. While the latter feature is not essential, since simulations and experiments of BAN stacks are carried out with the sink collecting data produced by the sensors (i.e., direct tree routing), the greater robustness of Pairwise likely leads to achieve actually better results in simulation, especially in terms of data packet delivery ratio.

Table 4.1 reports the list of specific parameter settings for the simulated BAN stack. It is worth pointing out that Pairwise's parameters have no direct correspondence with those of NSafeLinks, so they have been fixed in order to make the (Pairwise) protocol complete the routing tree building procedure. The pretty long initialization time dedicated to the routing tree building procedure is not actually (CPU) real time: Omnet++ is a discrete time simulator, so it "jumps" to initialization end time as soon as the actual procedure is completed.

Table 4.1 - Set of BAN stack's simulation parameters.

Model	Parameter	Value	Unit	Description
Routing (Pairwise)	Number of routing tree	1		Only one route to the sink
	Sink address	0		Sink S1
	Iteration delay	240	sec	Period between two tree-building procedures initiated by the sink
	Decision delay	60	sec	Time to wait before taking a decision on the favourite parent
	Ack delay	60	sec	Time to wait for a joining ACK from a favourite child
	Association delay	60	sec	Time to wait for a joining ACK from a favourite parent
	Max number of updates	4		Number of tree-building procedures to perform
	Metric	2		Hop count based metric
	Metric threshold	0.05		Ignore links with BER > 0.05
	Header length	8	byte	Length of the routing header
MAC(IEEE 802.15.4)	Busy RSSI	-97	dBm	Level at which the medium is considered busy
	Header length	120	bit	Length of the MAC header, including timestamp and CRC
	Queue length	100		Size of the MAC transmission queue
	Ack length	88	bit	Length of the ACK's MAC header, including timestamp and CRC
	Minimum backoff exponent	7		
	Maximum backoff exponent	11		

The HC stack is modelled and simulated with two different configurations. While the routing layer is implemented by the Gradient Routing protocol in both cases, the MAC layer is implemented by the 802.15.4MAC in the first case and by WiseMAC in the second. Both of these models relate to the implementations used on the hardware platforms tested during the project, namely Crossbow's TelosB motes and BSN platforms. The second configuration actually represents the WASP solution targeting mesh networks with mobile nodes.

The parameters that influence the behaviour of the system (and its model) in the two configurations are described in Table 4.2 and Table 4.3.

The communication is modelled by a so called path loss model. This model relates the reliability of the link between nodes with the distance between them.

Supporting protocols, such as time synchronization, are not implemented in the model, either because we have a global time or the protocols do not influence the metrics we want to determine with the models.

Table 4.2 - Set of HC stack's simulation parameters with WiseMAC

Model	Parameter	Value	Unit	Description
Routing (Gradient)	Number of routing tree	1		Only one route to the sink
	Sink address 1	0		Sink S1
	Sink address 2	10		Sink S2
	Max bc entries	100		Maximum number of broadcast messages
	Bc delete time	100	sec	Time after which broadcast messages are deleted
	Time to live	150	sec	Default time to live for a packet
	Update interval	100	sec	Time after which a new gradient routing tree is set up
	Header length	24	byte	Length of the routing header
WiseMAC	Max Tx attempts	3		Max numer of retransmission
	CCA threshold	-85	dBm	RSSI threshold for busy channel
	Rx threshold	-85	dBm	RSSI threshold for ongoing transmission
	Sampling interval	200	msec	Time interval between two channel pollings
	Minimum backoff exponent	7		
	Maximum backoff exponent	11		

Table 4.3 - Set of HC stack's simulation parameters with IEEE 802.15.4MAC

Model	Parameter	Value	Unit	Description
Routing (Gradient)	Number of routing tree	1		Only one route to the sink
	Sink address 1	0		Sink S1
	Sink address 2	10		Sink S2
	Max bc entries	100		Maximum number of broadcast messages
	Bc delete time	100	sec	Time after which broadcast messages are deleted
	Time to live	150	sec	Default time to live for a packet
	Update interval	100	sec	Time after which a new gradient routing tree is set up
	Header length	24	byte	Length of the routing header
MAC(IEEE 802.15.4)	Busy RSSI	-97	dBm	Level at which the medium is considered busy
	Header length	120	bit	Length of the MAC header, including timestamp and CRC
	Queue length	100		Size of the MAC transmission queue
	Ack length	88	bit	Length of the ACK's MAC header, including timestamp and CRC
	Minimum backoff exponent	7		
	Maximum backoff exponent	11		

4.3 Simulation parameters

The behaviour of a system (and of its model) is a function of its parameters. Based on the information required in the previous sections and the analysis of the hardware behaviour the following tables can be constructed, in which the parameters of the system (and of its model) are described. Beside the parameters in Table 4.1, Table 4.2 and Table 4.3, that are generic parameters for each scenario and protocol stack, there are specific parameter settings for each use case, whose summary tables are reported in the related sections.

4.3.1 Settings for Road Tunnel Scenario

We refer to a tunnel section of 300 meters and a network composed by eleven nodes. Figure 4.11 shows the deployment of nodes inside the tunnel section.

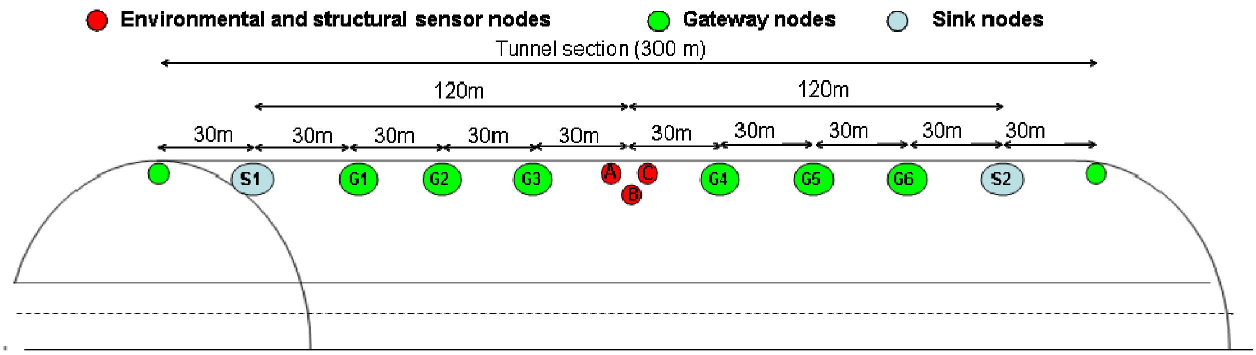


Figure 4.11 - Deployment of nodes inside the tunnel.

- A tunnel section of 300 m is considered.
- No obstacles (like machinery or equipment) are considered amongst the nodes of the network infrastructure.
- Red spots represent the environmental and structural sensor nodes. In normal conditions, they send packet with different transmission rates.
    - Node A sends unicast packets to S1 every 4 second, starting from $t_0 = 0 + \Delta t_0$ sec (Δt_0 is a random number with uniform distribution; first packet generation time is always chosen uniformly to avoid systematic collision), then $t1 = t_0 + 4$ sec, $t2 = t_0 + 8$ sec and so on.

- Node C sends unicast packets to S2 every 4 second, with 2 sec delays in respect to Node A. Therefore it starts sending at $t_0 = 2 + \Delta t_0$ sec, then $t_1 = t_0 + 4$ sec, $t_2 = t_0 + 8$ sec and so on.
- Node B transmits unicast to S1 every 10 seconds a burst of 1 packet each 200 msec, for, at least, a period of 2 seconds.

- Green spots represent the forwarder nodes; they have simply to forward the messages sent by the sensor nodes to the sinks, consistently with the rules and mechanisms provided by the specific routing algorithms.
- Light blue spots represent the sink nodes. They collect messages sent by sensor nodes, calculate the end to end latency and the delivery ratio.
- Before sensor nodes start sending packets to the sinks, an initial stage for the setting of all the protocols parameters is considered. This stage is strictly protocol stack dependent and includes only those stack parameters to be changed at run time.

Such a testbed should guarantee a sufficiently challenging environment for our trials. The presence of three aligned sensor nodes, which are, practically, equidistant from the sinks, and that, in the predefined instants, send periodically the same type of messages, should show us, in a long period observation, which are the average times related to the processing of the MAC algorithms.

At the same time, the contemporary presence for each sensor nodes of at least two paths towards the sinks composed by the same numbers of neighbour forwarders should offer the possibility to compare the times requested for addressing and routing.

Table 4.4 reports the list of generic parameter settings for the simulated use cases.

Table 4.4 - Generic parameters for the Road Tunnel Scenario.

Model	Parameter	Value	Unit	Description
General	Simulation time	700	sec	
	Repeat	60		Number of simulation runs to average
Network	X-size	300	m	
	Y-size	8	m	
	Number of hosts	11		
Channel	Carrier frequency	2.4	GHz	
	Path loss model	Free space		Signal power decay with distance
	Alpha	3		Path loss exponent
Radio	Transmission power	1	mW	0 dBm
	Sensitivity	-94	dBm	
	PHY header lenght	48	bit	IEEE 802.15.4 PHY
	Sleep power consumption	0,063	µW	0,021µA x 3V
	Reception power consumption	56,4	mW	18,8mA x 3V
	Transmission power consumption	52,2	mW	17,4mA @ -25dBm x 3V
	Rx-Tx turnoround power consumption	56,4	mW	18,8mA x 3V
	Tx-Rx turnoround power consumption	56,4	mW	18,8mA x 3V

Model	Parameter	Value	Unit	Description
Node A	Address	4		
	X-coordinate	145	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter #1	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	60	sec	Time to wait before generation of first packet
	Destination sink address	0		
Node B	Address	5		
	X-coordinate	150	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
	Application traffic type	Burst		Two different duty cycles
	Application traffic parameter #1	0.2	sec	First duty cycle
	Application traffic parameter #2	2	sec	Period of first duty cycle
	Application traffic parameter #3	10	sec	Second duty cycle
	Number of packets to be sent	600		1 packet every 200msec for a period of 2sec, repeated
	Application payload length	6	byte	
	Initialization time	60	sec	Time to wait before generation of first packet
	Destination sink address	0		
Node C	Address	6		
	X-coordinate	155	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter #1	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	62	sec	Time to wait before generation of first packet (delayed by
	Destination sink address	10		
Node G1	Address	1		
	X-coordinate	60	m	
	Y-coordinate	4	m	
	Speed	0	m/s	Static node
Node G2	Address	2		
	X-coordinate	90	m	
	Y-coordinate	4	m	
	Speed	0	m/s	Static node
Node G3	Address	3		
	X-coordinate	120	m	
	Y-coordinate	4	m	
	Speed	0	m/s	Static node
Node G4	Address	7		
	X-coordinate	180	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
Node G5	Address	8		
	X-coordinate	210	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
Node G6	Address	9		
	X-coordinate	240	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
Node S1	Address	0		
	X-coordinate	30	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node
Node S2	Address	10		
	X-coordinate	270	m	
	Y-coordinate	4	m	
	Speed	0	m/sec	Static node

4.3.1.1 UC1 - Alarm from a sensor in a normal situation

Sensor node A detects an abnormal condition. The node must perform all the actions needed to ensure the reception of the alarm by the tunnel management centre through node S1, until the centre itself does not cancel the alarm condition. The network is working properly.

The list of actions that need to be performed in such a situation is described here below:

- Sensor node A issues an alarm message to sink S1;
- Once received the alarm message, S1 manages the request and forwards it to the Control Centre:
- Sensor node A continues sending messages to S1 with the sensed values of the abnormal parameter every 1 sec, until the management centre, through node S1, does not cancel the alarm condition;
- Once the alarm condition has been managed by the control centre (in our simulation we've considered after 10 sec), S1 sends to node A the message *Alarm withdrawn;*
- Once received the *Alarm withdrawn* from S1, A starts again transmitting with its predefined duty cycle.

Figure 4.12 depicts the message sequence flow listed above:

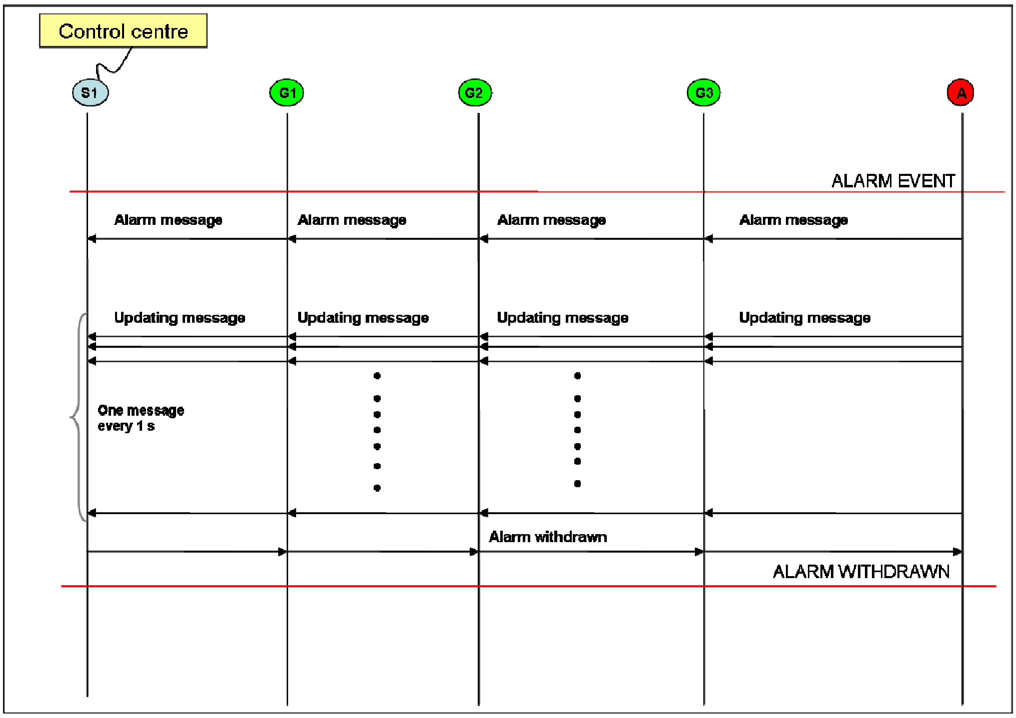


Figure 4.12 - Message sequence chart for UC1

To implement this case, a new type of message, ALARM_MESSAGE, has been introduced at application level and the possibility to set the alarm state for each node of the network. For this type of message, the destination sink can be set at application level and each sink is assigned by the control centre to reply to these messages in order to clear the alarm condition. The alarm condition generated by one of the node causes for such a node the interruption of the normal functioning, the interruption of the scheduled messages transmission. The normal functioning is restored again when the alarm condition has been withdrawn. To implement such scenario, besides the alarm state it is also possible to set the initialization time and the alarm message transmission frequency. At network layer, this type of message is managed like the others. At application layer, it is processed only by sink nodes; the other nodes have simply to forward the message towards sink.

4.3.1.2 UC2 - Alarm from a sensor when one or more forwarders are compromised

Sensor node A detects an abnormal condition. The node must perform all the actions needed to ensure the reception of the alarm by the tunnel management centre through node S1, until the centre itself does not cancel the alarm condition.

Unlike the previous use case, once the abnormal condition is detected by node A, the network is not properly working because one or more forwarders are compromised. In particular, the case when one of the forwarder nodes between sensor node A and the reference sink S1 is here considered. If the closest forwarder, used in normal conditions to reach the wired infrastructure is not operational, the WSN must take actions in order to try to communicate with the tunnel management centre using an alternative path.

The list of actions that need to be performed in such a situation is described here below:

- Sensor node A issues an alarm message addressed to sink S1;
- After several consecutive transmissions without acknowledgement from S1, the network readdresses the alarm message to S2 through forwarder nodes G4, G5 and G6;
- Sensor node A continues sending messages with the sensed values of the abnormal parameter every 1 sec, until the management centre, through node S2, does not cancel the alarm condition;
- Once the alarm condition has been managed by the control centre (in our simulation we've considered after 10 sec*)*, S2 sends to node A the message *Alarm withdrawn;*
- Once received the *Alarm withdrawn* from S2, A starts again transmitting with its predefined duty cycle.

Figure 4.13 depicts the message sequence flow listed above:

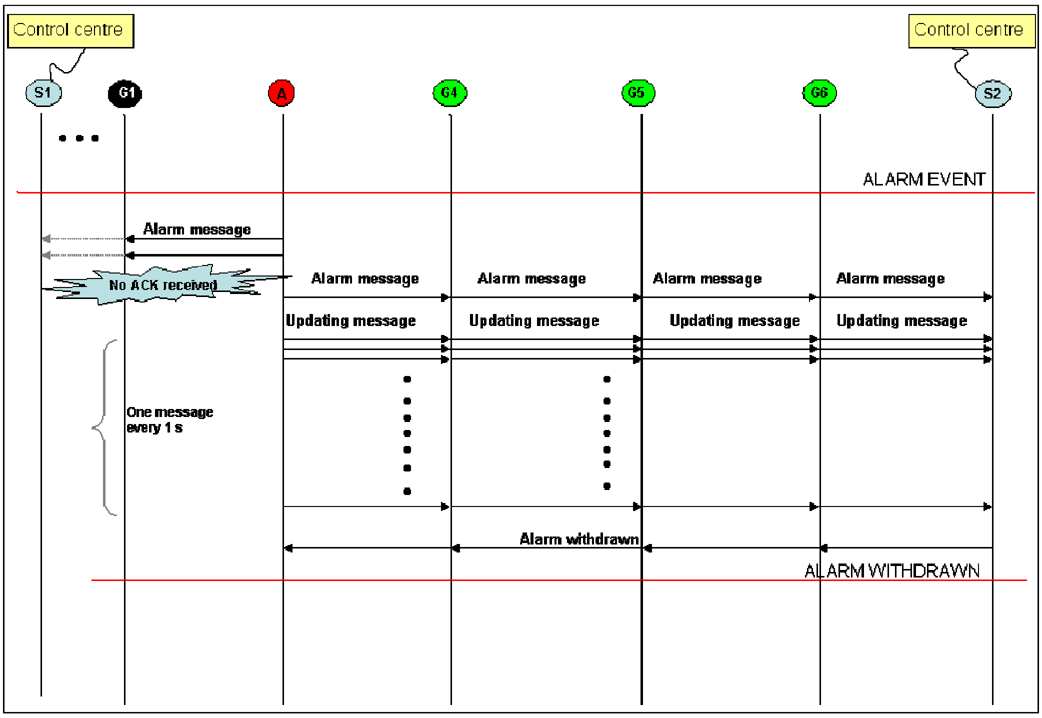


Figure 4.13 - Message sequence chart for UC2

The presence of a fault in the path between the node A and the sink S1 involves sending all packets generated at the application layer to the sink S2, as this is the only minimum distance path into network level to reach a sink. The sink S2, receiving ALARM_MESSAGE packets not addressed to it, stores the generation time. After 15 seconds from first generation time, the sink S2 runs the packet to the Control Centre and sends an Alarm Withdrawn to the node A.

4.3.1.3 UC3 - Periodic check of node status and battery level

The network should be able to periodically check the status of all the nodes in an autonomous fashion.

The list of actions needed to perform the periodic check is described here below:

- Each sink node Sx issues a check request message to the neighbouring nodes. Such a message is then forwarded by the forwarder nodes following the routing algorithm implemented by the protocol stack;
- Each sink node addresses its check request message to all the closest nodes (sensors and forwarders), from the two closest forwarders to the three sensor nodes;
- Once received the check request message, each node replies to the requesting sink with a reply message containing info about its status and battery level;
- Each sink continues broadcasting check request messages every 30 sec. Each reply is forwarded to the control centre;
- In case of missing reply from nodes, the control centre marks such a node as not functioning, reports and activates all the needed actions to restore the normal functioning;

Figure 4.14 depicts the message sequence flow listed above:

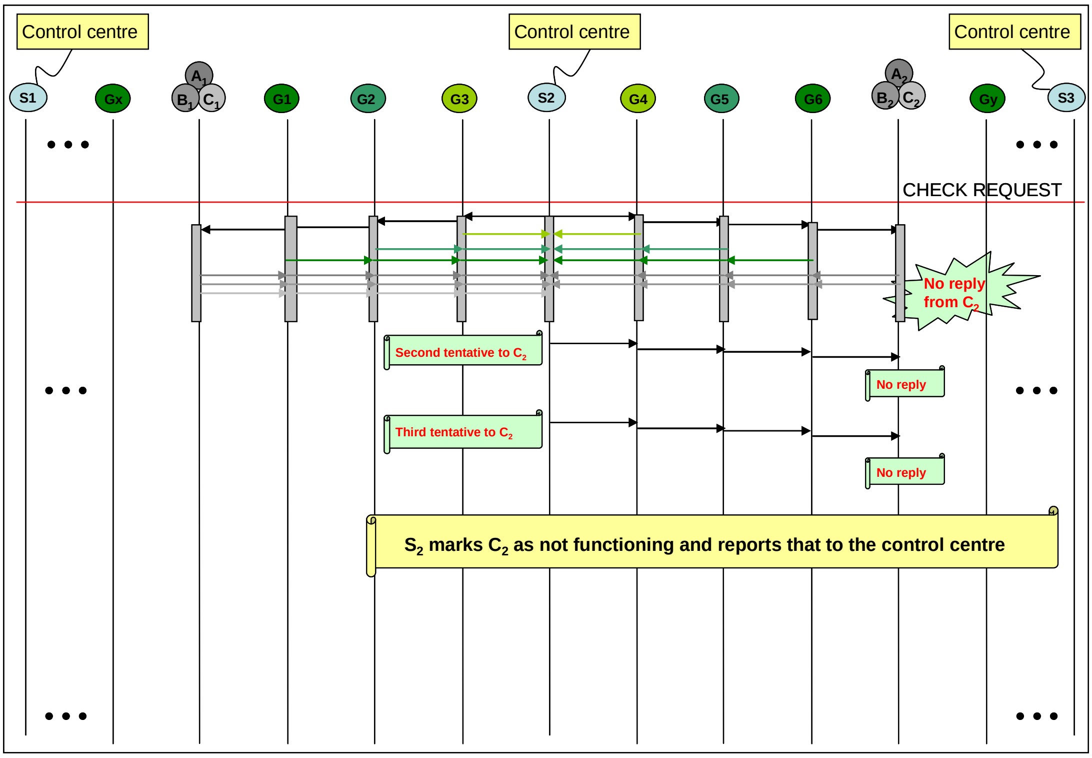


Figure 4.14 - Message sequence chart for UC3

From a programming point of view, a new message type, SERVICE_MESSAGE, has been introduced together with the possibility to program a sink node so that it can send messages of this new type. At network level, service messages are managed like the any other message and forwarded at application layer for each node. When a node receives a service message, it replies sending a message of the same type to the reference sink. Sinks then collect replies from each node and forwards to the upper layers toward the control centre.

4.3.1.4 UC4 - Check of nodes by the maintenance/emergency operators

Assistance vehicles should be able to perform a direct check on the status and the correct functioning of each node using a direct connection, while moving (slowly) along the tunnel. This procedure is used during the network deployment and for maintenance/emergency purposes. The vehicle speed is set equal to 20 Km/h (5.5 m/s).

The collection should be possible even if some part of the network is not operational, because of the emergency situation. This function should not interfere with the connection between the sensor nodes and the control centre.

The list of actions needed to perform the check is described here below:

- While moving along the tunnel, the vehicle node sends broadcast messages to the neighbouring nodes (sensors, sinks and forwarders) of the wireless network.
- One request message each 5 sec is broadcasted by the vehicle node moving along the tunnel.
- Once received the check request message, each node reacts to the request with a reply message containing info about its status and battery level.

Table 4.5 - Mobility Parameters of RT-UC4.

Model	Parameter	Value	Unit	Description
Mobility Node M1	X	300	m	x starting point of node (-1 = random)
	Y	7	m	y starting point of node (-1 = random)
	Speed	5.5	m/s	speed of the host
	Angle	180	degrees	angle of linear motion
	Accelerration	0	m/s2	acceleration of linear motion
	Updade Interval	0.1	s	update frequency

4.3.1.5 UC5 - Interaction between vehicles and infrastructure

Finally, the interaction between the private vehicles and the infrastructure of wireless network is tested. Here, a vehicle speed equal to 50 Km/h (14 m/s) is considered.

The list of actions needed to warrant the interaction is described below:

- While moving along the tunnel, the vehicle node sends multicast messages to the neighbouring sink nodes with info of speed, motor status and/or (flat) tire pressure.
- One message each 1 sec is sent by the vehicle node while moving along the tunnel.
- Once received the message, a sink reply with an ACK message;
- The vehicle node continues sending messages until it receives an ACK from a sink node.

Table 4.6 - Mobility Parameters of RT-UC5.

Model	Parameter	Value	Unit	Description
Mobility Mode M1	X	300	m	x starting point of node (-1 = random)
	Y	7	m	y starting point of node (-1 = random)
	Speed	14	m/s	speed of the host
	Angle	180	degrees	angle of linear motion
	Accelerration	0	m/s2	acceleration of linear motion
	Updade Interval	0.1	s	update frequency

4.3.2 Settings for In-vehicle sensor networks scenarios

4.3.2.1 Intra-vehicle network

We refer to a network composed by ten nodes. Figure 4.15 shows the deployment of nodes inside the car.

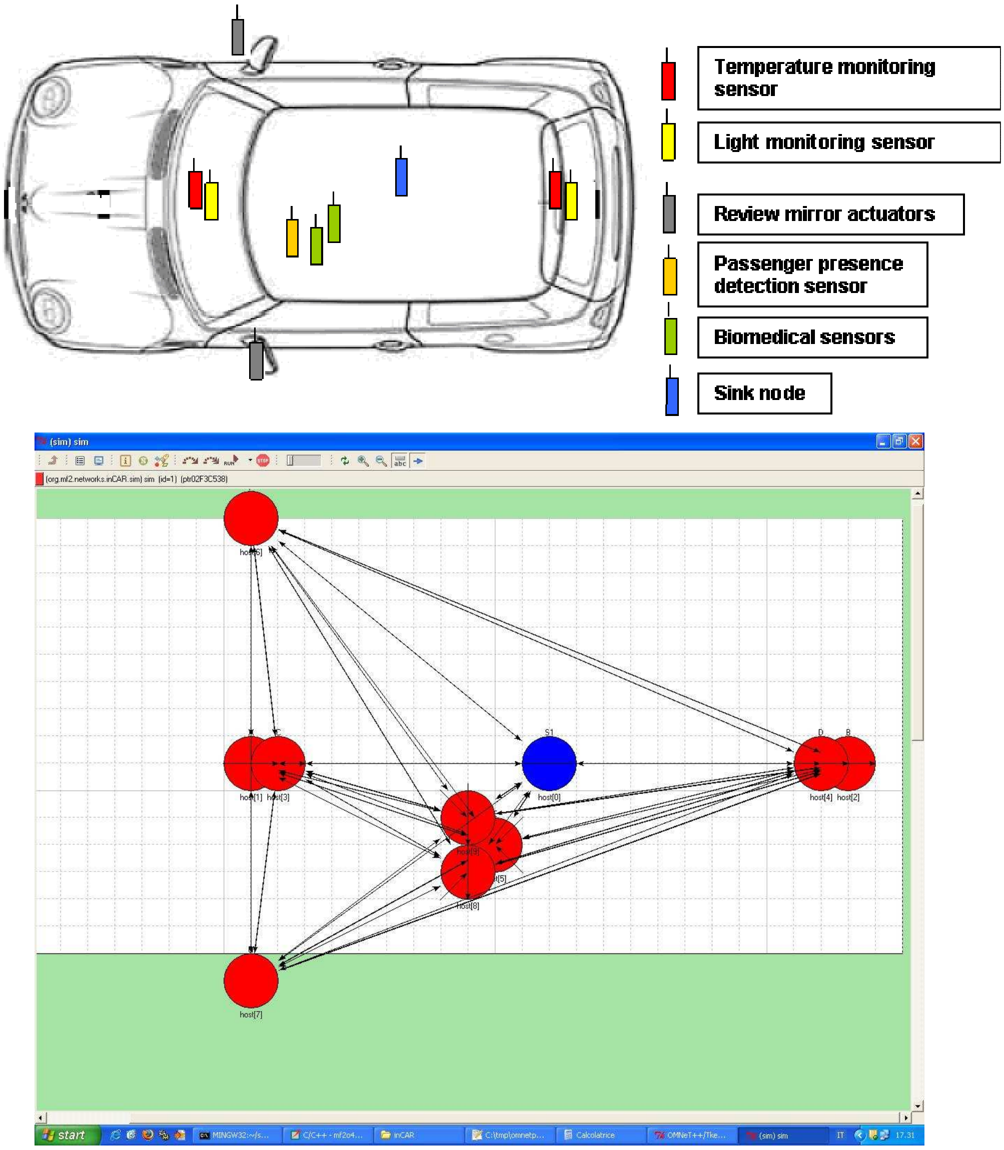


Figure 4.15 - Deployment of nodes within the vehicle compartment.

- A car compartment of 3.5 m x 1.6 m is considered.
- The communication channel is modelled considering possible absorptions and reflections of the radiofrequency signal, caused respectively by the presence of driver and passengers inside the vehicle and by the chassis.
- Each simulation test has duration equal to 1 hour, subdivided in 6 sessions with duration of 10 minutes each.
- A network composed by several wireless sensors and actuators, sending packets with different rates, is considered.

- Node S1 represents the sink node. It collects messages sent by sensor and actuators, calculates the end to end latency and the delivery ratio;
- Node A and B represent the internal temperature monitoring sensors. They periodically send unicast packets to the sink every 4 seconds, starting from $t_0 = 0 + \Delta t_0$ sec, then $t_1 = t_0 + 4$ sec, $t_2 = t_0 + 8$ sec and so on;
- Node C and D represent the internal light monitoring sensors. They periodically transmit unicast to the sink every 4 seconds, with 2 sec delays in respect to Node A and B. Therefore they start sending at $t_0 = 2 + \Delta t_0$ sec, then $t_1 = t_0 + 4$ sec, $t_2 = t_0 + 8$ sec and so on;
- Node E represents the passenger presence detection sensor. It sends unicast packets to the sink every 10 seconds a burst of 1 packet each 200 msec, for, at least, a period of 2 seconds.
- Node F and G represent the two rear view mirrors actuators. They send unicast packets to the sink every 10 seconds a burst of 1 packet each 200 msec, with 2 sec delays in respect to Node E.
- Node H and I represent two biomedical sensors. They send unicast packets to S1 every 20 seconds.

- Before sensor nodes start sending packets to the sinks, an initial stage for the setting of all the protocols parameters is considered. This stage is strictly protocol stack dependent and includes only those stack parameters to be changed at run time.
- Further tests could be carried out with different values of transmission rate in order to evaluate the respective variation in E2E latency and Data Packet Delivery ratio.

Tables 4.7 and Table 4.8 report, respectively, the list of generic parameter settings for the simulated use cases and the protocol stack parameters influencing the behaviour of the system (and its model).

Table 4.7 - Generic parameters for the In-vehicle sensor networks scenario.

Model	Parameter	Value	Unit	Description
General	Simulation time	36602	sec	Inizialization time + Host C's delay time + 10m test time
	Repeat	60		Number of simulation runs to average
Network	X-size	3.5	m	
	Y-size	1.6	m	
	Number of hosts	10		
Channel	Carrier frequency	2.4	GHz	
	Path loss model	Free space		Signal power decay with distance
	Alpha	5.1		Path loss exponent
Radio (TelosB)	Transmission power	1	µW	-29 dBm
	Sensitivity	-94	dBm	
	PHY header lenght	48	bit	IEEE 802.15.4 PHY
	Sleep power consumption	3	µW	1µA x 3V
	Reception power consumption	69	mW	23mA x 3V
	Transmission power consumption	29.4	mW	9.8mA @ -25dBm x 3V
	Rx-Tx turnoround power consumption	69	mW	23mA x 3V
	Tx-Rx turnoround power consumption	69	mW	23mA x 3V

Table 4.8 - Protocol stack parameter setting

Model	Parameter	Value	Unit	Description
Routing (Pairwise)	Number of routing tree	1		Only one route to the sink
	Sink address	0		Sink S1
	Iteration delay	240	s	Period between two tree-building procedures initiated by the sink
	Decision delay	60	s	Time to wait before taking a decision on the favorite parent
	Ack delay	60	s	Time to wait for a joining ACK from a favorite child
	Association delay	60	s	Time to wait for a joining ACK from a favorite parent
	Max number of updates	4		Number of tree-building procedures to perform
	Metric	2		Hop count based metric
	Metric threshold	0.05		Ignore links with BER > 0.05
	Header length	8	byte	Length of the routing header
MAC (IEEE802.15.4)	Busy RSSI	-97	dBm	Level at which the medium is considered busy
	Header length	120	bit	Length of the MAC header, including timestamp and CRC
	Queue length	100		Size of the MAC transmission queue
	Ack length	88	bit	Length of the ACK's MAC header, including timestamp and CRC
	Minimum backoff exponent	7		
	Maximum backoff exponent	11		

Tables below report the parameter settings for each node of the network.

Table 4.9 - Parameter settings for the sink node S1

Model	Parameter	Value	Unit	Description
Sink node S1	Address	0		
	X-coordinate	2	m	
	Y-coordinate	0.8	m	

Table 4.10 - Parameter settings for node A

Model	Parameter	Value	Unit	Description
Node A	Address	1		
	X-coordinate	1	m	
	Y-coordinate	0.8	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.11 - Parameter settings for node B

Model	Parameter	Value	Unit	Description
Node B	Address	2		
	X-coordinate	3	m	
	Y-coordinate	0.8	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.12 - Parameter settings for node C

Model	Parameter	Value	Unit	Description
Node C	Address	3		
	X-coordinate	1.1	m	
	Y-coordinate	0.8	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36002	sec	Time to wait before generation of first packet (delayed by 2sec w.r.t A and B)
	Destination sink address	0		

Table 4.13 - Parameter settings for node D

Model	Parameter	Value	Unit	Description
Node D	Address	4		
	X-coordinate	2.9	m	
	Y-coordinate	0.8	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36002	sec	Time to wait before generation of first packet (delayed by 2sec w.r.t A and B)
	Destination sink address	0		

Table 4.14 - Parameter settings for node E

Model	Parameter	Value	Unit	Description
Node E	Address	5		
	X-coordinate	1.8	m	
	Y-coordinate	1.1	m	
	Application traffic type	Burst		Two different duty cycles
	Application traffic parameter #1	0.2	sec	First duty cycle
	Number of packets to be sent	600		1 packet every 200msec for a period of 2sec, repeated
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.15 - Parameter settings for node F

Model	Parameter	Value	Unit	Description
Node F	Address	6		
	X-coordinate	1	m	
	Y-coordinate	0.5	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36002	sec	Time to wait before generation of first packet (delayed
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.16 - Parameter settings for node G

Model	Parameter	Value	Unit	Description
Node G	Address	7		
	X-coordinate	1	m	
	Y-coordinate	1.55	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	4	sec	Transmission delay time
	Number of packets to be sent	150		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36002	sec	Time to wait before generation of first packet (delayed
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.17 - Parameter settings for node H

Model	Parameter	Value	Unit	Description
Node H	Address	8		
	X-coordinate	1.7	m	
	Y-coordinate	1.2	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	20	sec	Transmission delay time
	Number of packets to be sent	30		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.18 - Parameter settings for node I

Model	Parameter	Value	Unit	Description
Node I	Address	9		
	X-coordinate	1.7	m	
	Y-coordinate	1	m	
	Application traffic type	Periodic		Two different duty cycles
	Application traffic parameter	20	sec	First duty cycle
	Number of packets to be sent	30		1 packet every 200msec for a period of 2sec, repeated
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Such a testbed should guarantee a sufficiently challenging environment for our trials. The presence of such type of sensor nodes, which are, practically, equidistant from the sinks, and that, in the predefined instants, send periodically the same type of messages, should show us, in a long period observation, which are the differences amongst the average times related to the processing of the MAC algorithms.

4.3.2.2 Tire Pressure Monitoring System

We refer to a network composed by five nodes, the four TPM sensors and the TPM module. Figure 4.16 shows the deployment of the TPMS network inside the vehicle.

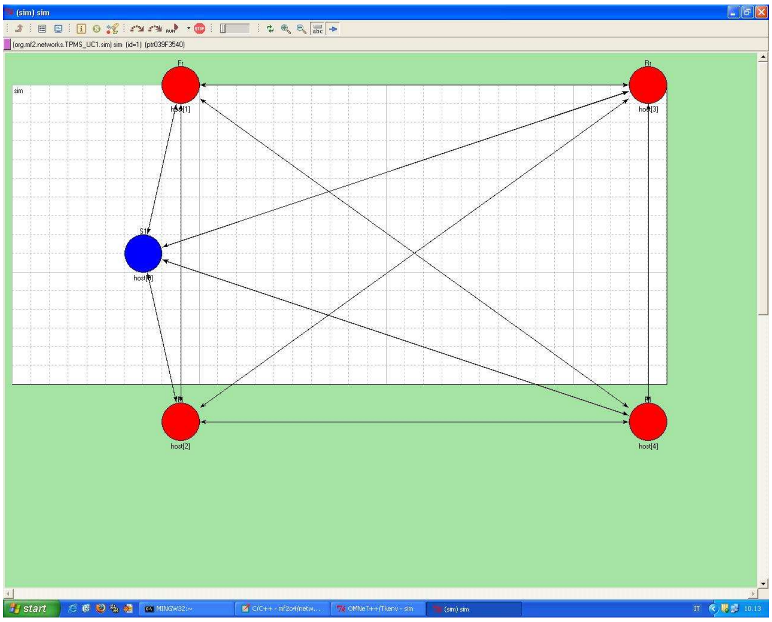


Figure 4.16 - Deployment of the TPMS network

- The communication channel is modelled considering possible signal reflections, caused by the shielding of the chassis.
- Node S1, implementing the TPM module, represents the sink node. It collects messages sent by sensors, calculates the end to end latency and the delivery ratio.
- Node Fl represents the front left TPM sensor
- Node Fr represents the front right TPM sensor
- Node Rl represents the rear left TPM sensor
- Node Rr represents the rear right TPM sensor

Table 4.19 reports the list of generic parameter settings for the simulated use cases.

Table 4.19 - Generic parameters for the In-vehicle sensor networks scenario.

Model	Parameter	Value	Unit	Description
General	Simulation time	42000	sec	Inizialization time + Host C's delay time + 10m test time
	Repeat	60		Number of simulation runs to average
Network	X-size	3.5	m	
	Y-size	1.6	m	
	Number of hosts	5		
Channel	Carrier frequency	2.4	GHz	
	Path loss model	Free space		Signal power decay with distance
	Alpha	5.1		Path loss exponent
Radio (TelosB)	Transmission power	1	µW	-29 dBm
	Sensitivity	-94	dBm	
	PHY header lenght	48	bit	IEEE 802.15.4 PHY
	Sleep power consumption	3	µW	1µA x 3V
	Reception power consumption	69	mW	23mA x 3V
	Transmission power consumption	29.4	mW	9.8mA @ -25dBm x 3V
	Rx-Tx turnoround power consumption	69	mW	23mA x 3V
	Tx-Rx turnoround power consumption	69	mW	23mA x 3V

The protocol stack parameters that influence the behaviour of the system (and its model) are described in Table 4.20.

Table 4.20 - Protocol stack parameter setting

Model	Parameter	Value	Unit	Description
Routing (Pairwise)	Number of routing tree	1		Only one route to the sink
	Sink address	0		Sink S1
	Iteration delay	240	s	Period between two tree-building procedures initiated by the sink
	Decision delay	60	s	Time to wait before taking a decision on the favorite parent
	Ack delay	60	s	Time to wait for a joining ACK from a favorite child
	Association delay	60	s	Time to wait for a joining ACK from a favorite parent
	Max number of updates	4		Number of tree-building procedures to perform
	Metric	2		Hop count based metric
	Metric threshold	0.05		Ignore links with BER > 0.05
	Header length	8	byte	Length of the routing header
MAC (IEEE802.15.4)	Busy RSSI	-97	dBm	Level at which the medium is considered busy
	Header length	120	bit	Length of the MAC header, including timestamp and CRC
	Queue length	100		Size of the MAC transmission queue
	Ack length	88	bit	Length of the ACK's MAC header, including timestamp and CRC
	Minimum backoff exponent	7		
	Maximum backoff exponent	11		

Such a testbed should guarantee a sufficiently challenging environment for our trials. The presence of four sensor nodes, which are practically, equidistant from the sinks, and that, in the predefined instants, send periodically the same type of messages, should show us, in a long period observation, which are the differences amongst the average times related to the processing of the MAC algorithms.

4.3.2.2.1 UC1 – Normal Mode - Nodes deployment

Each node periodically sends unicast packets to the sink every 60 seconds, starting from $t_0 = 0 + \Delta t_0$ sec, then $t_1 = 60 + t_0$ sec, $t_2 = 60 + t_1$ sec and so on.

Tables below report the parameter settings for each node of the network.

Table 4.21 - Parameter settings for sink node S1

Model	Parameter	Value	Unit	Description
Sink node S1	Address	0		
	X-coordinate	0.6	m	
	Y-coordinate	0.8	m	

Table 4.22 - Parameter settings for node Fl

Model	Parameter	Value	Unit	Description
Node Fr	Address	1		
	X-coordinate	0.8	m	
	Y-coordinate	0	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	60	sec	Transmission delay time
	Number of packets to be sent	100		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.23 - Parameter settings for node Fr

Model	Parameter	Value	Unit	Description
Node Fl	Address	2		
	X-coordinate	0.8	m	
	Y-coordinate	1.6	m	
	Application traffic type	Periodic		Two different duty cycles
	Application traffic parameter	60	sec	First duty cycle
	Number of packets to be sent	100		1 packet every 200msec for a period of 2sec, repeated
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.24 - Parameter settings for node Rl

Model	Parameter	Value	Unit	Description
Node Rr	Address	3		
	X-coordinate	3.1	m	
	Y-coordinate	0	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	60	sec	Transmission delay time
	Number of packets to be sent	100		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet (delayed
	Destination sink address	0		Single sink (S1) for BAN stack

Table 4.25 - Parameter settings for node Rr

Model	Parameter	Value	Unit	Description
Node Rl	Address	4		
	X-coordinate	3.1	m	
	Y-coordinate	1.6	m	
	Application traffic type	Periodic		Constant packet rate
	Application traffic parameter	60	sec	Transmission delay time
	Number of packets to be sent	100		1 packet every 4 sec for a period of 10 min
	Application payload length	6	byte	
	Initialization time	36000	sec	Time to wait before generation of first packet (delayed
	Destination sink address	0		Single sink (S1) for BAN stack

4.3.2.2.2 UC2 - Alert Mode

We refer to a network that works in normal mode and we consider these actions:

- The wheel sensors are in "normal mode";
- An alarm occurs;
- The TPM changes sensors from Normal mode to Alert mode working condition;
- In alert mode, tyre pressure and temperature data are transmitted every 4 seconds for 60 seconds duration.

4.4 Results

This chapter reports the results collected in simulation sessions. The realized stacks are assessed in respect with the metrics considered in the benchmark defined in [4.1] and here summarized in chapter 4.2.1.

4.4.1 Results analysis

A common approach is used to analyze the results from simulations. Based on statistical analysis as described in for example [4.5], the following sequence of actions is used to evaluate the metrics of the simulated WSNs.

- When a simulation run finishes, it produces a (.sca) file with the average x_i (i = 1..n) of each of the selected metrics. Considering the number of simulation runs fixed in Table 4.3, a set of n = 60 single averages are finally collected. Note that each single average is calculated not considering metric samples measured during the network initialization phase, when the application sources are still off.
- The expected value for each metric x_m, also said "sample mean", is given by averaging the single averages x_i according to the expression:

$$x_m = sum_{i=1..n}(x_i) / n = sum_{i=1..60}(x_i) / 60.$$

- The standard deviation of the single averages x_i with respect to the sample mean x_m, also said "sample standard deviation", is calculated according to the expression:

$$s = square\_root((sum_{i=1..n}(x_i - x_m)^2) / (n - 1)) = square\_root((sum_{i=1..60}(x_i - x_m)^2) / 59).$$

- For each metric to be estimated, the confidence interval at a fixed "unreliability" index alpha is then:

$$(x_m - s / square\_root(n) * z(alpha / 2), x_m + s / square\_root(n) * z(alpha / 2)) =$$

$$(x_m - s / square\_root(60) * z(alpha / 2), x_m + s / square\_root(60) * z(alpha / 2)].$$

- Based on alpha = 0.01, it is possible to say (see Table 2 in [4.5]) that the estimated value falls with 99% reliability degree into:

$$(x_m - s / square\_root (60) * z(0.005), x_m + s / square\_root (60) * z(0.005)) =$$

$$(x_m - s / square\_root (60) * 2.576, x_m + s / square\_root (60) * 2.576) \sim (x_m - s * 0.333, x_m + s * 0.333).$$

- For completeness, the minimum x_{min} and the maximum x_{max} values obtained for each metric are simply given by:

$$x_{min} = min_{i=1..n}(x_i) = min_{i=1..60}(x_i)$$

$$x_{max} = max_{i=1..n}(x_i) = max_{i=1..60}(x_i).$$

The way of sampling the selected metrics in time, in order to have final x_i averages for the i-th run, is hereunder described:

- Power consumption: when the initialization phase or the entire simulation ends, every node writes out the energy consumed by the radio until then (i.e., respectively, energyUsedAtInit and energyUsed both expressed in Joule) to the aforementioned .sca file; to evaluate the energy consumed on average in the network, it is needed to (read from the file and) subtract the sum of energyUsedAtInit values from the sum of energyUsed values, and divide the result by the number of nodes. To have a power value in mW, the final energy value in Joule is to be divided by the reference time period in seconds (i.e., simulation time minus initialization time) and multiplied by one thousand.
- Data packet delivery ratio: after sending a data packet, every sensor node increases a counter of its sent packets (i.e., nbPacketsSent); after receiving a data packet, the sink node increases a counter of its received packets (i.e., nbPacketsReceived); when simulation ends, the counters' values are written out to the aforementioned .sca file; to evaluate the network goodput, it is needed to (read from the file and) divide the nbPacketsReceived of the sink by the sum of nbPacketsSent of every sensor.

- End-to-end latency: when the sink receives a data packet, it calculates the packet's delivery latency as difference between the arrival time and the creation time (both stamped on the packet itself as control information) and updates the average of the collected latencies for the packet's originator (i.e., latency[i], where 'i' is a sensor's address); when simulation ends, the averaged latencies for each sensor are written out to the aforementioned .sca file; then, it is needed to (read from the file and) average the single per-node latencies to have a per-simulation-run average value.

4.4.2 In-vehicle scenarios

The following tables report the results obtained with the simulations described in Ch. 4.3.2.

Table 4.26 - Simulation results for Intra-vehicle network (alpha = 5.1)

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	61,251	61,653	61,452	61,422	61,482	0,05
Data packet delivery ratio (%)	100	100	100	100	100	0,00
End-to-end latency (s)	0,035	0,055	0,042	0,04	0,043	3,57

Table 4.27 - Simulation results for Intra-vehicle network (alpha = 2)

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	61,251	61,656	61,452	61,422	61,482	0,05
Data packet delivery ratio (%)	100	100	100	100	100	0,00
End-to-end latency (s)	0,0356	0,052	0,042	0,041	0,044	3,57

CI means "confidence interval" and CI_Error is the ratio between the confidence interval's radius (CI_Max - CI_Min) / 2 and the mean value.

Tables 4.26 and 4.27 report the results for the three selected metric in respect to different values of the path loss exponent alpha. In the study of wireless communications, path loss can be represented by the path loss exponent, whose value is normally in the range of 2 to 4 (where 2 is for propagation in free space, 4 is for relatively lossy environments and for the case of full specular reflection from the earth surface—the so called flat-earth model). In some environments, such as buildings, stadiums and other indoor environments, the path loss exponent can reach values in the range of 4 to 6. On the other hand, a tunnel may act as a waveguide, resulting in a path loss exponent less than 2.

The collected values prove the very limited impact of such a parameter in the tested scenario. The interaction amongst nodes inside the car compartment seems to be almost immune from absorptions and reflections due to presence of driver and passengers and for the presence of metal objects.

Table 4.28 - Simulation results for Intra-vehicle network (enhanced transmission rate)

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	61,382	61,721	61,558	61,532	61,586	0,04
Data packet delivery ratio (%)	100	100	100	100	100	0,00
End-to-end latency (s)	0,048	0,111	0,065	0,061	0,068	5,38

Results reported in Table 4.28 are instead obtained with an enhanced transmission rate of 1pkt per second. In respect to the previous cases (one packet every four seconds) the transmission rate is therefore quadruplicated. As we can see by the collected values, this causes a not negligible increase of the resulting latencies, up to few dozens of msec. This is ascribable to the delay introduced by the MAC layer to separate transmissions contending the channel (backoff delay). In connection with a transmission rate increase, there is an increase of application traffic in the network per time unit and consequently an enhanced contention level of the channel.

802.15.4MAC guarantees the delivery of all the transmitted packets (this is confirmed by the values collected for Data packet delivery ratio) but with a corresponding delay in delivery.

These first three cases prove the full reliability of the channel with 100% of data packet delivery ratio.

Variations of path loss exponent and transmission rate don't cause meaningful changes in power consumption, as better explained below. For data forwarding, every node consumed around 37 Joule that is 61.5 mW drawn per time unit through 602s of complete session. Considering 3V of supply voltage, the corresponding current absorption is close to 20.5 mA (61.5 mW / 3 V), such as to say that every node alternates continuously between transmitting and receiving, without sleeping. If a node was powered by two AAA batteries (1.5V, 1250 mAh), it would sustain the said consumption for approximately 61 hours (1250 mAh / 20.5 mA), that is a bit more than 2.5 days.

Table 4.29 and Table 4.30 report simulation results for the TPMS use case in respect with two different transmission rates: 1 packet every 60 second for the Normal Mode and 1 packet every 4 seconds for the Alert Mode. Despite this not negligible difference in transmission rate values, the three metrics don't change significantly. Just a not negligible deterioration of the minimum data packet delivery ratio can be pointed out. This is obviously caused by the increased contention level of the channel. In respect to the previous case (in-vehicle case under the same value of transmission rate [1pkt every 4 seconds]), here the contention level is tightened up by the fact that tyre sensors transmit simultaneously. Here the MAC layer attempts to retransmit packet after contention up to a predefined number of possible retransmissions; after which packets are discarded without any further tentative. Nevertheless it doesn't affect the average value, still close to 100%.

Table 4.29 - Simulation results for TPMS – Normal Mode

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	61,445	61,966	61,683	61,65	61,716	0,05
Data packet delivery ratio (%)	99,75	100	99,996	99,985	100	0,01
End-to-end latency (s)	0,04	0,057	0,044	0,043	0,044	1,14

Table 4.30 - Simulation results for TPMS – Alert Mode

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	61,496	61,951	61,678	61,645	61,71	0,05
Data packet delivery ratio (%)	77,25	100	99,62	98,643	100,599	0,98
End-to-end latency (s)	0,039	0,056	0,043	0,042	0,044	2,33

As a general remark, the adopted solution for the protocol stack guarantees more than 99% delivery on average, which is less than 1% of data packet loss, a very low latency, in the order of tens of milliseconds, resulting in a very reliable configuration for the tested applications. This is surely favoured by the low transmission rates considered in our trials, reducing the influence of collisions, back-offs and retransmissions.

On the contrary, collected results highlight high values of power consumption on average, close to 62 mW for both the cases. Considering the nominal values for the Transmission Power Consumption and the Reception Power Consumption, respectively 29 mW and 69 mW, as reported in Table 4.19, such a result depicts an unbalanced radio duty-cycle reflecting the high reception activity of nodes in the network. The high value is obviously due to the fact that 802.15.4 doesn't implement power saving techniques. This causes a high current absorption independently by the transmission rate, as the radio module is always active. This results in a very low autonomy of the node, to values clearly not acceptable for this type of applications.

4.4.3 Road tunnel scenario

The following tables report the collected from the simulations described in Ch 4.3.1.

Table 4.31 reports values collected for the three metrics in normal condition with no alarm, no service operator, no mobile nodes and no out of order forwarders.

Table 4.31 - Simulation results for the Road Tunnel scenario in normal condition

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	56,391	56,394	56,393	56,393	56,393	0,00
Data packet delivery ratio (%)	78,667	100	97,563	95,9	99,226	1,70
End-to-end latency (s)	0,052	0,069	0,057	0,056	0,056	0,00

Table 4.32 reports the results for UC1, when an alarm occurs in the network. Comparing the values of each metric with the previous case, there are no meaningful deviations from the results in normal functioning.

Table 4.32 - Simulation results for RT-UC1

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	56,39	56,394	56,393	56,392	56,393	0,00
Data packet delivery ratio (%)	88,778	100	99,457	98,821	100	0,59
End-to-end latency (s)	0,052	0,065	0,056	0,055	0,057	1,79

Table 4.33 reports values obtained for UC2 when during an alarm condition, one forwarder is compromised. The goodput is here referred to sink node 10, calculated as the ratio between the number of packets sent by the three sensors toward the sink node 10 and the number of packets received by the same node following the alternative path.

Table 4.33 - Simulation results for RT-UC2

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	56,385	56,394	56,393	56,392	56,394	0,00
Data packet delivery ratio (%)	94,555	99,222	98,204	97,911	98,498	0,30
End-to-end latency (s)	0,053	0,08	0,056	0,055	0,058	2,68

From a programming point of view, the fault of a forwarder node can be set at application layer inhibiting the sending and receiving of any kind of message from and to such a node. At network layer, the gradient routing warrants the forwarding toward the closet sink anyway.

Table 4.34 reports results for UC3 during the periodic check of network nodes.

Table 4.34 - Simulation results for RT-UC3

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	56,391	56,394	56,393	56,393	56,393	0,00
Data packet delivery ratio (%)	78,778	100	97,8	96,102	99,45	1,71
End-to-end latency (s)	0,051	0,077	0,056	0,055	0,057	1,79

Table 4.35 provides values obtained for UC4 when a moving operator interrogates the network sending SERVICE CONTIDION MESSAGES packets in order to collect functioning status information from each node. From a programming point of view, the behaviour is the same as in UC3.

Table 4.35 - Simulation results for RT-UC4

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	56,391	56,394	56,393	56,393	56,393	0,00
Data packet delivery ratio (%)	78,444	100	97,596	95,925	99,268	1,71
End-to-end latency (s)	0,052	0,069	0,056	0,055	0,056	0,89

Finally, Table 4.36 reports values collected for UC5, where the mobile vehicle node that interacts with the network infrastructure providing information on its status and vehicle's speed.

Table 4.36 - Simulation results for RT-UC5

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	56,391	56,394	56,393	56,393	56,393	0,00
Data packet delivery ratio (%)	75,082	99,012	97,386	95,987	98,785	1,44
End-to-end latency (s)	0,031	0,052	0,041	0,039	0,043	4,88

Host is here provided by Linear_Mobility modules to manage and set mobility. From the network point of view, the mobile node and the information it provides, are managed like for the other sensors. Such a node doesn't have a reference sink and sends broadcast packets. Such messages are forwarded through the network toward the nodes that have the lower gradients (hop distances).

Results demonstrate a substantial invariance of the three metrics in the simulated use cases. This is especially true for power consumption (around 56 mW on average) and data packet delivery ratio, ranging from minimum values never below the 90% to maximum values of 99% and more, for average values of around 98%. Values of latency are basically the same for the first four cases while a not negligible reduction of around 50msec is experienced in UC5. This can be justified by the fact that the vehicle node can reach each sink node while moving along the tunnel without delays introduced by hops.

The adopted stack solution is able to provide a data packet delivery ratio greater than 96%, with peaks of 99.555% also in alarm condition. Few packet losses are observed and they can be retraced to channel congestion. Additional cross-layer optimization should be applied in order to reduce channel occupancy. A way to do that could be limiting the number of retransmissions. When the Gradient protocol receives a packet, it checks its sending queue if there are any packets with the same identification number waiting to be transmitted. If there is any packet with the duplicated packet ID, the transmission of this packet should be cancelled to avoid unnecessary packet transmission.

Among the data packets that are received by the sink, it is experienced a delivery latency of 120ms on average, which is 40ms per hop: note that each sensor is three hops far from the reference sink. The most part of transmission delay is likely due to the high channel contention and the resulting pretty long backoff delay preceding the first transmission attempt.

As regards the power consumption, every node consumes for data forwarding around 34 Joule that is 56 mW drawn per time unit through 602s of complete session. Considering 3V of supply voltage, the corresponding current absorption is close to 19 mA (56 mW / 3 V). If a node was powered by two AA batteries (1.5V, 1250 mAh), it would sustain the said consumption for approximately 66 hours (1250 mAh / 19 mA), that is around 2.75 days.

In short, these results point out a quite acceptable reliability of the network in term of latency and delivery ratio but an even high current absorption, especially considering nodes sustained by batteries. Further improvement in performances could be achieved for all the metrics with even more customized cross layer optimizations and protocol selection but as regards the energy autonomy, future enhancement in energy saving should be anyway coupled with energy accumulation techniques and energy harvesting to ensure an unfailing and self regenerating power supply.

As anticipated in Chapter 4.2.3, the same use cases have been simulated also with a second configuration of the HC stack, composed by WiseMAC and Gradient routing, which is actually the WASP solution targeting mesh networks with mobile nodes.

WiseMAC is an ultra low power MAC protocol. Such a protocol tries to reduce the power consumption by regularly switching off the radio most of the time, minimizing the energy wastage in idle listening.

It uses a variable length preamble which is dynamically selected to minimize the power consumption. The preamble length duration is based on the neighbour's sleep schedule transmitted at the end of each data transmission in the acknowledgement message. Each node maintains a table containing sleep schedule information of its neighbouring nodes. WiseMAC does not require strict global clock synchronization and it does not fix a common wakeup schedules for all the nodes in the network. However, the neighbouring nodes know the schedules of each other and exchange data in their common active periods. The decentralized sleep-listen scheduling results in uncoordinated wakeups of neighbouring nodes that adds latency. In broadcast transmission, as in the case of gradient routing, this becomes more significant when the nodes in different sleep cycles buffer the broadcasted packets and then deliver it. This way the packets are broadcasted many times resulting in added latency and percentage of packet loss.

As for the first configuration based on 802.15.4MAC, results demonstrate a substantial invariance of the three metrics in the simulated use cases. We therefore report here below just the values collected for the three metrics in normal condition with no alarm, no service operator, no mobile nodes and no out of order forwarders.

Table 4.37 - Simulation results for the Road Tunnel in normal condition with second configuration

Metric	Min	Max	Mean	CI_Min	CI_Max	CI_Error (%)
Power Consumption (mW)	6,281	8,235	7,351	7,201	7,501	2,04
Data packet delivery ratio (%)	41,833	74,556	56,488	54,6	58,376	3,34
End-to-end latency (s)	10,687	95,813	56,048	50,484	61,612	9,93

Results reported in the table highlight that while this new configuration has a positive influence on the overall power consumption, it has a negative influence on the end-to-end latency and on the delivery ratio. The improvement in power consumption is more than considerable with resulting energy autonomy of around 21 days for nodes powered by two AAA batteries.

The increase in latency is ascribable to collisions, back-offs and retransmissions due to the high channel occupancy and contention due to broadcast transmissions imposed by Gradient. The resulting back-offs cycles determine for sink nodes and forwarders, when acting as receivers, a misalignment of the original sleep schedule and consequent misalignment in wakeups with neighbouring nodes. This again adds latency. The greater is the misalignment the greater the latency increase. Nodes are unable to send packets at the rate they are produced. Consequently the buffer is quickly filled and the latency is dominated by the buffer residence time. If we compare the simulation results of the previous configuration, for which the radio is always on, we can observe this behaviour.

At the same time, the higher is the packet queue, the greater the probability of successful retransmission and then the lower the probability that they are lost. Therefore the buffer size has to be established considering this trade-off.

If we look at the data delivery ratio obtained from simulation we see that only around 57% of the messages arrives at the sink on average. This is much less than the 80.15.4-based solution which approaches a 100% data packet delivery ratio.

As a general remark, this solution provides a good improvement in energy efficiency, although still far from what requested by the application requirements, but this at the expense of an overall worsening of the overall reliability of the system, both in terms of latency and delivery ratio.

4.5 Benchmarking

4.5.1 Motivation

Modern vehicles incorporate tens of sensors to provide information such as temperature, air quality, tire pressure, distance to nearby objects, etc., to the electronics control units (ECUs). The ECUs in the vehicles utilize the sensor information for various control functions and applications. In the current architecture, the sensors in the vehicle are connected to the ECUs via physical wires. The wiring harness in a car today could have up to 4000 parts, weigh as much as 40 kilograms and contain more than 1900 wires for up to 4 kilometres in length [4.8]. As the number of electronic components keeps increasing with the development of new features in the vehicles, the number of in-car sensors could be more than a few hundred in the near future. The physical wires connecting the ECUs and the sensors could become problematic due to the following reasons:

- Wires are expensive. The physical connecting wires are usually shielded so they can be heat and interference resistant. The shielded copper wires in the vehicles could become a major cost component as the number of sensors increase. Moreover, it requires a significant amount of engineering effort, and hence cost, to design the layout of those wires in the vehicle. The effort needs to be repeated for every different model of vehicle.
- Wires are heavy. Wires and wiring harness are among the heaviest components in the vehicle. The weight of the wires could have a large impact on the fuel efficiency as the number of in-car sensors increase in the near future.
- Wires are restrictive. There are several locations in the vehicle where sensors cannot be deployed when using the current wired sensors, e.g., steering wheels, tires, and windshields. In addition, it requires significant effort to add a new sensor to or change the location of a sensor from the existing design; the layout of the wires needs to be modified.

It is thus imperative to create a new open architecture to support communications between sensors and ECUs. To address the aforementioned issues, many researchers are trying to design an in-car wireless sensor network, which might replace at least part of the physical wires with wireless connections. In such a network, the sensors have wireless communication capability and broadcast packets which contain sensor information to the base station(s) periodically. The base station(s) then relay the sensor information to the ECUs for further processing. The main constraints of such a solution are:

- Cost. One of the major motivations of building such an in-car wireless sensor network is to cut down the cost of the expensive shielded connecting wires and it would not make sense to deploy such a system if it is much more expensive. Therefore, the designs of the sensor node and the base station must be as simple as possible, and all of the following challenges need to be addressed in a cost effective manner.
- Longevity. If the sensor nodes still need to be attached to a power supply via physical wires while the communication is wireless, it defeats the purpose of having an in-car wireless sensor network. Hence, the sensor node needs to be either passive, i.e., obtain the power from the electro-magnetic wave from the base station, or active, i.e., has an onboard battery. In either case, the energy available for the sensor node is highly constrained and the operation of the wireless sensor nodes, including radio transmission and reception, needs to be highly energy efficient. The lifetime of a sensor node should be in line with the designed lifetime of other components in the vehicle which is typically more than 5 years.
- *Communication Reliability.* Certain sensor information is crucial to the operation of the vehicle and needs to be delivered to the ECUs with a high success rate. The high channel loss and time-varying nature of the underlying in-car wireless channels could result in a

higher packet loss rate than when using physical wires, which may require additional system design to compensate.

- *Network capacity.* Although the effective data rate of each individual sensor node is usually only up to a few *Kb/s*, as the number of sensor nodes increase in the near future the overall network load could still reach a few hundred *Kb/s*. Due to the multipath-rich in-car wireless channels and hence the low coherence bandwidth for some locations in the vehicle, the network capacity might not be sufficient to support the operation of certain sensor nodes.
- *Heterogeneous sensors.* Different from most of other wireless sensor networks, the requirements and purpose of the sensors in the vehicle are considerably different. The importance and priority of the sensor packets sent by different sensors are very different and needs to be prioritized; some need to be delivered to the ECUs before the others do. The packet sending interval of different sensors could range from 6.25 *ms* to several seconds. It is therefore necessary to adopt a design which can accommodate these heterogeneous characteristics of the sensors.

The main challenges and issues to be investigated for a comparison with legacy solutions are:

- Which is the level of reliability that can be provided by the in-car wireless channels in terms of packet reception rate?
- Which is the level of reliability that can be provided by the in-car wireless channels in terms of maximum packet delay?
- Which is the order of magnitude for power consumption that can be provided by wireless nodes with current hardware platforms and networking strategies?

The following sections provide some references and theoretical background necessary for answering these questions.

4.5.2 Application for tyre pressure monitoring system

An application example in the category of virtual wire is a tire pressure monitoring system. Because it is impractical to replace the sensors or their batteries between tire changes, it is required that the sensor batteries last at least three years and preferably more. This puts significant constraints on the power consumption of the electronic components and requires power management capabilities.

The data that needs to be communicated (the measured tire pressure) is, in most cases, only a few bits in size. This information is transmitted from about 1 to 10 minutes under no-alarm conditions. Unless there is a fast loss of pressure, the message latency is not of significant concern. In case of a sudden pressure loss, the central control unit, should be notified immediately; in which case, the power consumption is not of concern because most likely the tire has to be consequently replaced.

Extreme automotive environmental conditions and the metallic structure of the car complicate the RF design. In addition, the shape of the rim has a significant impact on the radiation pattern from the wireless sensor. To overcome this issue, repeater devices, which should not add a significant cost to the system, can be added to the network to increase communication reliability. Of course, an implicit advantage not present in actual legacy solutions is the possibility to have two-way communications with the sensor, thus increasing the reliability of the system.

In the following section, a compatibility analysis between WASP specifications and TPMS requirements is conducted. This is done by first highlighting the features in WASP platform which can help satisfying the requirements of TPMS. Following that, possible weaknesses which may potentially limit the use of WASP in this application are introduced. It is important to note that the analysis conducted in this section focuses only on the operational features of WASP and TPMS, assuming the existence of a reliable channel, and disregarding the external effects (resulting from rotation, heating, breaking….etc.) which can influence the robustness of that channel.

4.5.2.1 TPMS Requirements

The power efficiency issue is one of a prime concern in the TPMS application. Generally, some of the operational techniques that help achieve low average power consumption are reduction of the amount of data transmitted, reduction of the transceiver duty cycle and implementing strict power management mechanisms such as power-down and sleep modes. WASP solution as the IEEE 802.15.4/Zigbee technologies is specified with a wide range of low power features at physical and higher levels which can help achieve the low power consumption requirement of the TPMS application; they can be summarised as follows:

- The general operational power-saving feature warranted by the low duty cycle operation. Typical applications are expected to run with low duty cycles (under 1%). It is worth noting, however, that if the warm-up time which wakes the device up from its sleep mode is high, significant power may be unnecessarily consumed. Warm-up time can be dominated by the settling of transients in the signal path; especially the integrated active channel filters.
- Wideband techniques, such as the Direct Sequence Spread Spectrum (DSSS) employed in WASP, have an advantage that their wide channel filters have inherently short settling times. With their greater channel spacing, DSSS frequency synthesisers may also employ higher frequency references, reducing lock time.
- The offset-QPSK with half-sine pulse shaping modulation selected for the 2.4GHz PHY allows the use of simple, low-cost components and constitutes low demand on the linear performance of the power amplifier, reducing the need for potential high power consumption.
- Specifications for receiver blocking (defined as the receiver ability to receive a wanted signal at its assigned channel frequency in the presence of an unwanted interferer on frequencies other than those of the adjacent channels) is reduced, allowing lower active power consumption of the receiver front-end. RF Modules like the Chipcon CC2420 of BSN nodes adopted in WASP provide very good adjacent, alternate and co channel rejection, image frequency suppression and blocking properties.
- Output power specifications are reduced. Platform like BSN must be capable of transmitting -3dBm (as a lowest maximum), but can operate at lower power (as low as -32dBm) if acceptable by the application and the environment.

Some additional features that can be beneficial to the requirements of the TPMS application include:

- Direct Sequence Spread Spectrum (DSSS) and modulation techniques implemented by the most used RF modules, assure high bandwidth efficiency and increased security.
- Transmission band can be up to 10 metres and addressing scheme can support up to 255 active nodes per network sink. This means a TPMS realised using WASP can be used in a variety of vehicles ranging from small personal cars into large trucks.
- Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA) protocol allows to avoid wasteful collisions when multiple simultaneous transmissions occur. This exactly fits the requirement to avoid collisions if one or more sensor modules attempted to transmit to the central module simultaneously.

This analysis doesn't exhaust the subject but based on the aforementioned considerations, it can be concluded that IEEE 802.15.4-based technologies, like WASP, have the potential to realise an efficient TPMS. Off course improvements are needed before such application can become a physical reality. For instance, it is evident that some of the above listed power reduction techniques can lead to severe effects on the quality of the transmission of signals. One of the goals will be then to find a way by which to improve the signal quality but without affecting the power consumption.

4.5.3 In-car Wireless Sensor Networks

Currently, sensors and Electronic Control Unit (ECU) in a car communicate over serial data busses and are connected with physical wires. Replacing signal wires, for some of the sensor/switches in selected subsystems, with wireless communications is subject to maintaining the same order of latency and reliability of communication currently achieved by wired connectivity. This is of paramount importance for each electronic control unit (ECU) to get its input on time, execute the control algorithm and deliver its monitor/control output on time.

The main problem is which level of reliability can be provided by the in-car wireless channels in terms of packet reception rate and maximum packet delay. To be more specific, it is essential to determine whether the requirements of packet reception rate and maximum packet delay, which are imposed by the specifications of the sensors, can be fulfilled by the characteristics of the in-car wireless channels.

There are a lot of papers in literature describing surveys and experimentations for the deployment of wireless sensor networks inside a car [4.8][4.9][4.10][4.11]. They provide refined statistical analysis of results achieved in real tests with different configurations. In [4.9], ElBatt, Saraydar, Ames and Talty provide a formula to calculate the latency that could be achieved in communicating a sample measurement from a sensor/switch node to the ECU in a star topology network IEEE 802.15.4-based.

$$\boxed{\text{802.15.4\_UL\_Latency} = 15.36 * \left\lceil \frac{N}{7} \right\rceil + 0.544 \quad \text{msec} \qquad N > 1}$$

Two observations can be made based on the result above. First, the latency grows linearly with the number of nodes. Second, 802.15.4 can not support smaller than 15.9 msec latency for any star network of any size. This, in turn, implies that 802.15.4 MAC can not support in-vehicle subsystems with sub millisecond latency requirements.

The result is of great interest, since it directly determines the class of wireless applications that can be realistically implemented inside a vehicle: depending on the level of reliability targeted for different applications, the formula provides an important indication on which sensors can be accessed wirelessly and for which sensors this might be more troublesome. It points to a possible hybrid in-car sensor network where some sensors are accessed wirelessly while others are accessed via wires.

In a typical modern vehicle, the most demanding sensor application will require a latency of approximately less than 1 msec with throughput of 12 kbps. Further, the network needs to support about 15 sensors with these requirements. The least demanding sensor application requires a latency of approximately 50 msec with data throughput rate of 5 kbps and needs to support about 20 of these types of devices. Hence, the requirements cover a wide spectrum.

Sensors and actuators forming the intra-vehicle network simulated with the WASP protocol stack architecture, mainly belong to the second class of sensor application. Values for latency obtained from simulations range from 43 ms to 48 ms on average. They are surely higher than the theoretical values provided by the formula (for 11 nodes it provides a latency of around 27 ms) but lower than the boundary imposed from requirements for low demanding sensor application. We can therefore conclude that the tested protocol stack architecture can be considered as a reliable solution to be implemented in an in-vehicle wireless sensor network, for all those applications where the data are not being used to make challenging actuations and mission critical decisions.

4.5.4 Power consumption of currently available platforms

Several IEEE 802.15.4/ZigBee compliant chips are currently available in the market from different manufacturers. Table 4.38 provides a comparison between the current consumption and supply voltage of various transceivers in the 2.4GHz band, using data obtained from their respective data sheets. As can be seen, power consumption varies significantly.

Table 4.38 - Power consumption of commercial 802.15.4-based chips

	Supply Voltage	RX-mode Current (mA)	TX-Mode Current (mA)	Power Output (dBm)	RX-Power Consumption (mW)	TX-Power Consumption (mW)
Freescale (MC13191)	2.0 to 3.4 (2.7 Typ)	30	37	-27 to +4	60 to 102	74 to 125.5
Atmel (AT86RF230)	1.8 to 3.6	16	17 (Max TX 3dBm)	-17 to +3	28.8 to 57.6	30.6 to 61.2
Ember (EM 250)	1.7 to 1.9	28	28 (Max TX 3dBm)	-32 to+3	47.6 to 53.2	47.6 to 53.2
TI/Chipcon (CC2420)	2.1 to 3.6	19.7	17.4	-25 to 0	41.4 to 70.9	36.5 to 62.6

Calculations using these data show that for the current ratings it is not possible to build a TPMS system because an AAA battery would last for 2 years, assuming moderate use. Bearing in mind that some TPMS regulations require the in-tyre module battery to last 10 years and that an AAA battery is probably too heavy to be placed in a tyre, the need for reduced power consumption is clear.

On the contrary, the same values provide good openings for intra-vehicle applications, where such theoretical estimations for battery duration are quite acceptable.

4.5.5 Cost/benefit budget

Aside from technical considerations, the issues relating to the economic and business side of this concept play a key role for the future of WSN inside the car.

An in-depth and quantitative discussion of these issues is outside the scope of this document, but a few comments can be made. Vehicle manufacturing/assembly costs include the deployment of wires for a vast number of sensors from point to point. Being able to eliminate this wiring translates directly in the elimination of a step on the assembly line. However, if the wireless solution cost is greater than the cost of the wire and wiring, then the technology appeal is moot: there simply is no business case.
As things stand today, unfortunately, wire replacement technologies and wireless sensors do not present a strong business case. This is precisely because off-the-shelf wireless sensor/switch hardware, for whatever protocol standard being considered, costs at least tens of dollars per node, at best. This level of cost is unsustainable as a wire replacement alternative. But there is hope. If standards can be established for the hardware components and software interface for the radios, for the sensors and subsystems, then parts vendors can integrate the technology into their process flow, thereby allowing volume manufacturing, and as has been the case historically for all silicon-based microelectronics products, silicon costs then will eventually approach zero per unit device.

4.6 Road Management lessons learned

The applicability of WASP solutions has been tested and assessed by performing simulations for some selected scenarios which are recognized, at present, as the most promising on the short to medium term.

We consider networks composed from several wireless sensor nodes enabling sensing and monitoring in a vehicle environment and inside a long tunnel.

The BAN stack has been adopted for simulations of the in-vehicles network applications whereas the HC stack has been used for the tunnel network. Different simulation models have been defined and tested for all the selected use cases, in order to evaluate network performances (e.g. end-to-end latency), reliability (delivery ratio), power consumption (node lifetime) and overall node activity.

The analysis is not focussed on very specific sensor applications. The design of the networks considers a generic deployment of nodes aimed to assess the sustainability of the overall node activity rather than to tune minutely the configuration parameters of a particular sensor node. The sensors referred in the trial have been selected just to set some domain-specific requirements. The number of nodes in the network, their deployment, the transmission rates and the interaction rules, have been set as generic as possible in order to take into account the majority of possible cases and sensors.

Results coming from the simulations demonstrate that the WASP solution, adopted for the protocol stack targeting Body Area Networks and here experimented with in-vehicle applications, guarantees very good network performances and reliability, with values of delivery ratio very close to 100% (99% delivery on average, which is less than 1% of data packet loss) and very low latency, in the order of tens of milliseconds. Collected results depict nevertheless high values of power consumption on average, resulting in a very limited autonomy of the nodes, especially considering devices sustained by batteries.

The same overall behaviour results from the simulation of a network of wireless sensors for data replication within a tunnel, with a protocol stack composed by IEEE802.15.4MAC and Gradient Routing. Simulations of the same scenario based on a protocol stack composed by WiseMAC with Gradient, which actually represents the WASP solution targeting mesh networks with mobile nodes, provide on the contrary a good improvement in energy efficiency, although still far from what requested by the application requirements, but this at the expense of an overall worsening of the overall reliability of the system, both in terms of latency and delivery ratio.

We can therefore conclude that the tested WASP protocol stack architecture can be considered as a dependable solution to implement in-vehicle wireless sensor network, surely in terms of packet reception rate but also in terms of maximum packet delay, especially for all those applications where the data are not being used to make challenging actuations and mission critical decisions with sub millisecond latency requirements.

On the contrary, power consumption is at present the biggest fundamental bottleneck toward the introduction of such a technology in cars and automotive applications. This represents one of the biggest questions to be addressed and further investigated by researchers in the next year. Further improvement in performances could be achieved for all the metrics with even more customized cross layer optimizations and protocol selection but as regards the energy autonomy, future enhancement in energy saving should be anyway coupled with energy accumulation techniques and energy harvesting to ensure an unfailing and self regenerating power supply.

5 Requirement fulfilment

At the end of the prototype development phase it is good to have feedback to the original requirements. In D6.2 the generic and application specific requirements were described. Prototype development took place on the EC1 scenario ‘Activities of daily living using wearable/ambient sensors’ and HC2 scenario ‘Detection of health problems with focus on claw health and locomotion’.

In the WP6 meeting in June 2010 in London the table of generic requirements was scored. The yellow scores in Table 5.1 are the generic requirements that were picked up during prototyping. These results show that the prototyping included a vast majority of the generic requirements, which is a promising result. The table gives no quantitative results of the requirement fulfilment.

Table 5.1 - Generic requirement fulfilment during prototyping

Number	Requirement code	Requirement Name	Node hardware[a]	Software platform[a]	Networking protocols[a]	Services[a]	Security[a]	Power management[a]	Quality of Service[a]	RM1[b]	RM2[b]	RM3[b]	RM4[b]	EC1[b]	EC2[b]	EC3[b]	EC4[b]	HC1[b]	HC2[b]	HC3[b]
1	R-GN-004	Water-proof nodes	+	–	–	–	–	–	–	M	M	M	M					M	M	M
2	R-GN-041	Biocompatibility	+	–	–	–	–	–	–								M			M
3	R-GN-002	Rechargeable batteries	+	–	–	–	–	+	–	M	M	M	M	M	M	M	M			
4	R-GN-003	Temperature range	+	–	–	–	–	+	–	M	M	M	M					M	M	M
5	R-GN-009	Environment-proof	+	–	–	–	–	+	–	M	M	M						M	M	M
6	R-GN-022	Dimensions and weight of nodes	+	–	–	–	–	+	+			X		M	M	M	M	M	M	M
7	R-GN-028	Sensor standardization	+	–	–	–	+	–	–					M	M	M	M	S	M	M
8	R-GN-023	Legislation compliance	+	+	–	–	–	–	–			M		M	M	M	M			
9	R-GN-055	Accuracy of measurements	+	+	–	–	–	–	–											M
10	R-GN-026	Number of parameters per node	+	+	+	–	–	+	–					M	M	M	M	M	M	M
11	R-GN-007	Multi-hop communication	+	+	+	–	–	+	+	M	M			M	M	M	M	M	M	M
12	R-GN-047	Network programming/rebooting	+	+	–	+	–	–	–					M	M	M	M			
13	R-GN-054	Multiple applications possibility	+	+	–	+	–	–	–									W	C	C
14	R-GN-042	Calibration	+	+	–	+	–	–	+								S			C
15	R-GN-001	Low level of battery notification	+	+	–	+	–	+	+	M	M	M	M	M	M	M	M	C	S	S
16	R-GN-013	Failure detection	+	+	–	+	–	+	+	M	M	S		M	M	M	M	C	S	S
17	R-GN-030	Backup possibility	+	+	+	+	–	–	–					M	M	M	M		C	M
18	R-GN-019	Automatic alarming based on threshold detection	+	+	+	+	–	–	+			M		M	M	M	M	M	M	
19	R-GN-033	Network address claiming	+	+	+	+	–	–	+					S	S	S	S	S	S	M
20	R-GN-039	Data compression	+	+	+	+	–	–	+					S					C	M
21	R-GN-048	Mobility recording	+	+	+	+	–	–	+									C		M
22	R-GN-016	Location awareness	+	+	+	+	–	+	–		S			S	S	S	S	M	M	M
23	R-GN-044	Device autonomy	+	+	+	+	–	+	–	M	M									
24	R-GN-008	Synchronization	+	+	+	+	–	+	+	M	C	C		M	M	M	M	W	M	M
25	R-GN-014	Data storage	+	+	+	+	–	+	+		C			M	M	M	M			M
26	R-GN-012	Dynamic network configuration	+	+	+	+	–	+	+	M	M	M		M	M	M	M	S	S	M
27	R-GN-021	Ability to reconfigure sampling rate/characteristics	+	+	+	+	–	+	+			X		M	M	M	M		M	S
28	R-GN-037	Activity measurements	+	+	+	+	–	+	+					S					M	W
29	R-GN-020	Layered Structure of Network	+	+	+	+	+	–	–			S		S	S	S	S		M	M
30	R-GN-027	Sensor identification	+	+	+	+	+	–	+					M	M	M	M	W	W	M
31	R-GN-018	Sensor data availability	+	+	+	+	+	+	+	M	M			M	M	M	X			
32	R-GN-035	Reliable end-to-end transport	+	+	+	+	+	+	+		M			M	M	M	M	W	M	S
33	R-GN-034	Interaction with input/output devices	+	+	+	+	+	+	+					S	S	S	S	M	M	M
34	R-GN-040	Availability of discovery service	+	+	+	+	+	+	+					M	M	M	M		M	M
35	R-GN-029	Node attachment	+	–	+	–	–	–	+					M	M	M	M	M	M	M
36	R-GN-036	Wireless frequency	+	–	+	–	–	+	+					M	M	M	M			M
37	R-GN-049	Communication within BSN	+	–	+	+	–	+	–										M	M
38	R-GN-053	Availability of communication data	+	–	+	+	–	+	–					M	M	M	M			
39	R-GN-005	Sensor node lifetime	+	–	+	+	–	+	+	S	S	S	S				S	M	M	M
40	R-GN-045	Failure-proof system	+	–	+	+	–	+	+	M	M	M								
41	R-GN-024	Integrity of sensor data	+	–	+	+	+	–	+	S	C			M	M	M	M	M	S	M
42	R-GN-025	Limited number of body-sensors	+	–	+	+	+	–	+					M	M	M	M	M	M	M
43	R-GN-052	Sensor Network Audit Trail	+	–	–	+	–	–	–					S	S	S	S			
44	R-GN-043	Ease of diagnostic	+	–	–	+	–	–	+	M	M	M								
45	R-GN-006	Ease of use	+	–	–	+	–	–	+	M	M			M	M	M	M			

46	R-GN-046	Ease of maintenance	+	−	−	+	−	−	+		S	S								
47	R-GN-050	Tests procedures and devices	+	−	−	+	−	−	+		M									
48	R-GN-051	Network reconfiguration ability	+	−	−	+	−	−	+					M	M	M	M			
49	R-GN-017	Sensor thresholds configurability	−	+	+	+	−	−	+		S	S		M	M	M	M	M		
50	R-GN-031	Generating summary reports	−	+	+	+	−	−	+					M	M	M	M		M	
51	R-GN-032	Confidentiality of sensor data	−	+	+	+	+	−	+		M			M	M	M	M	M	M	S
52	R-GN-010	Security/authentication	−	+	+	+	+	+	+	S	C			M	M	M	M			X
53	R-GN-038	Monitoring based on different variables	−	−	+	+	−	+	+					C					C	
54	R-GN-011	Access control/authorization	−	−	+	+	+	−	−	M				M	M	M	M			M
55	R-GN-015	Compatibility with existing infrastructure	−	−	−	+	−	−	+		M			M	M	M	M			

Looking to the generic requirements it can be stated that the requirements on ‘Biocompatibility, Rechargeable batteries, Legislation compliance, Device autonomy, Access control/ authorization’ were not picked up during the prototype development. These requirements will become more evident when commercial applications will be developed.

Secondly it can be stated that during the iterative process the focus was on the development of the components and the integration towards the tests in the test bed environment. These were the driving factors. The different generic requirements as such were not the driving factor for the development, but they definitely played an important role in the unconsciousness of the developers. They new that it has to fit the requirements and looking to the results of table 5.1 one can see that indeed almost all generic requirements were integrated in the WASP concept.

The WASP concept proved to be in more ways a generic concept. The components so far that had to be made real application specific were the on-nod classifiers and algorithms and the packaging of the nodes.

6 Concluding remarks

After having discussed the setup of the prototyping, simulation and testing of the WASP concept within the three business areas of elderly care, herd control and automotive, we close this deliverable with some concluding remarks from the application perspective. For this the ‘promises’ of WASP, summarized in chapter 1, will be used.

Smart wireless multi sensor network (WSN) with wireless access to ‘Internet environment’

The WASP concept is more or less using a layered construction. In the upper levels form the routers to the EIC the connection to the Internet and the use of Internet based services is used. In the experiments in Elderly Care and Herd Control also this Internet connection proved to be valuable. Testing and uploading of new software could be done remotely. Also the integration of the TelMed system and the GPRS forwarder showed the integration in the Internet environment.

State based triggering of sensors and WSN.

The chosen scenarios for elderly care and herd control were quite challenging for the WASP development. Although at first site they look ‘simple’ they had enough similarities and differences to drive the developing process. For the state based triggering work has been performed on the algorithm development and classification methods of the accelerometer and ECG based sensors. Algorithms could be developed and tested in such a way that they could be programmed and uploaded with the ECA and uDSSP programming tools. The algorithms showed to be robust and efficient enough. It is also shown that the WASP concept can be used to put ‘all’ application functionality into one program, as can be seen in the Herd Control prototype, or to put the application functionality in different modules, as was shown in the Elderly Care scenarios.

Smart energy, security, accuracy, storage and processing strategies within WSN and through services available in the neighbourhood.

One of the big challenges from the beginning was the energy use of the battery powered WASP nodes. In the iterative process several steps were wade to limit the energy use. From application point of view the development of the on-node algorithms reduced data transfer a lot. In the network we have seen that the introduction of the BAN and the HC stack, replacing a very basic temporary stack to get integration going, did initially not result in an energy-efficient, robust and stable network. A significant development effort has been devoted to remove the various sources of instabilities and in deliverable D8.4 two case examples are included to illustrate the complexity of debugging constrained devices with rather limited debug facilities. In the end, we did manage to remove the embedded coding stability issues and managed to get low-power WiseMAC integrated.

Concerning security, light-weight solutions for both Data Access Control (on application layer) and packet encryption (on physical layer) have been implemented and validated in stand-alone tests. Integration in the actual test beds has been not pursued to avoid further complication of the on-going activies. However, in the prototypes we did implement and tested trustworthiness assessment of sensor data into the EIC.

The storage of the data is solved on node and backend (enterprise) level. For this, data formats used in both Elderly Care and Herd Control were aligned and allowed for central sensor data storage and handling by the EIC. Also, a prototypical farm information system build on top of the EIC has been demonstrated as further described in D8.4.

Moving objects becomes central in real time environment (time and location).

The developed prototypes are able to deal with moving objects whether it is a human a cow or a car. Based on the generic requirements and the prototype description, one can state that the need for real 'real time' was not present. Reaction times of seconds and minutes were acceptable in the scenarios. This is fully exploited in the chosen solutions. Further exploiting these non-real time functionality will give more room for energy efficient network creation. Some simulation experiments were done to explore this.
With regard to the location awareness the integration of the HouseVibe sensor can be mentioned. It is integrated and one can tell whether a person in a certain room is moving or standing still. In the Herd Control experiments the location awareness is build in by using the traditional RSSI values of the moving nodes and the static forwarders. Integration of DV-distance in the HC stack has been successfully demonstrated but not tested in the final integrated dairy farm demo. Instead, a DV Distance calibration procedure has been developed that greatly reduces the calibration time during actual deployment.

User friendly application development environment.

In the start of WASP the idea was to come from 'sand to application'. In the work performed of WASP this indeed became reality by integration of the whole chain. During test bed development, choices were made in line with the envisioned test bed planning but integration meetings needed to be flexible based on actual availability and maturity of components. A step-wise introduction of components in to the test beds was crucial for success and provide the required flexibility. Nevertheless, integration of test bed prototypes and successful deployment of technology trials proved to be a very time-consuming activity that left little time for long-time application (data collection) trials.

Concerning the gap between developers and researchers within the WASP consortium, we clearly achieved major improvements. During the project a lot of discussions were needed to properly understand each other and use the same terminology. The contribution of the architecture team and the formulation of the WASP architecture pictures were quite helpful. In the last year, we have seen that the use of the SVN repository helped a lot in integration components from various partners. At this moment, the application users of WUR and ICL work really together with the software developers located throughout Europe and actually do part of the software work themselves. Also the use of the simulation tools provided in the project was picked up by the automotive application builders. So one can state that the essence of the WASP concept contributes to the closing of the gap between the researchers and the application builders.

A wider collection of lessons learned can be found in deliverable D8.4 where also main lessons from other workpackets are collected.

References

[2.1] Kenney, W. L., Munce, T. A. (2003). Invited Review: Aging and human temperature regulation. *J. Appl. Physiol.* 95: 2598-2603

[2.2] WASP project, Deliverable D6.3 ”Test plan and description of test beds”, September 2008.

[2.3] Physical activity and cardiovascular health. NIH Consensus Development Panel on Physical Activity and Cardiovascular Health. JAMA 1996 Jul 17; 276(3):241-6.

[2.4] Wannamethee SG, Shaper AG. Physical activity in the prevention of cardiovascular disease. Sports Med 2001; 31(2):101-114.

[2.5] Go AS, Hylek EM, Philips KA, Chang Y, Henault LE, Selby JV, Singer DE. Prevalence of diagnosed atrial fibrillation in adults: national implications for rhythm management and stroke prevention: the AnTicoagulation and Risk Factors in Atrial Fibrillation (ATRIA) study. JAMA 2001. 285, 2370–2375.

[2.6] Chobanian AV, et al. The seventh report of the joint national committee on prevention, detection, evaluation, and treatment of high blood pressure: the JNC 7 report. JAMA 2003. 289, 2560–2572.

[2.7] A.G. Society, G. Society, A.A. of, and O.S.P.o.F. Prevention, "Guideline for the Prevention of Falls in Older Persons". Journal of the American Geriatrics Society, 2001. 49(5): p. 664-672.

[2.8] D. Keskin, P. Borman, M. Ersöz, A. Kurtaran, H. Bodur, and M. Akyüz, "The Risk Factors Related to Falling in Elderly Females".Geriatric Nursing. 29(1): p. 58-63.

[2.9] E. Mitty and S. Flores, "Fall Prevention in Assisted Living: Assessment and Strategies". Geriatric Nursing. 28(6): p. 349-357.

[2.10] Real time pervasive monitoring for post operative care. B. Lo, L. Atallah, O. Aziz, M. ElHelw, A. Darzi, GZ. Yang in BSN 2007, Aachen, Germany.

[2.11] W.A.H. Engelese and C. Zeelenberg: “A single scan algorithm for QRS detection and feature extraction”, IEEE Comput. Card., Long Beach: IEEE Computer Society, 1979, pp. 37-42.

[2.12] Aliot, E., Nitzschè, R., Ripart, A. Arrhythmia detection by dual-chamber implantable cardioverter defibrillator. A review of current algorithms. Europace 6, pp. 273 – 286, 2004.

[2.13] HL7 Annotated ECG (aECG): http://wiki.hl7.org/index.php?title=Product_RR_aECG
For a detailed description of the aECG format see
http://www.amps-llc.com/UsefulDocs/aECG_Implementation_Guide.pdf

[2.14] Regulated Clinical Research Information Management committee: mission and overview accessible through: http://www.hl7.org/Special/committees/rcrim/index.cfm

[3.1] Lokhorst, C., , L. Atallah, P. Hogewerf, R. de Mol, E. Frumento, L. Campelli, M. Mazzu, R. Verhoeven, A. Laube, A. Barroso, M. Schuster, L. Thiem, F. Ingelrest, J. Lukkien & M. Bennebroek, 2008b. WASP, Test plan and description of test beds, Deliverable 6.3.

[3.2] R.M. de Mol, R.J.H. Lammers, J.C.A.M. Pompe, A.H. Ipema & P.H. Hogewerf, 2009. Recording and analysis of locomotion in dairy cows with 3D accelerometers. In: C. Lokhorst & P.W.G. Groot Koerkamp (eds.) Precision livestock farming '09. Papers presented at the 4th European Conference on Precision Livestock Farming, Wageningen, The Netherlands, 6-8 July 2009, pp 333-341.

[3.3] R.M. de Mol, E.J.B. Bleumer, P.H. Hogewerf and A.H.Ipema. Recording of dairy cow behaviour with wireless accelerometers. In: C. Lokhorst & P.W.G. Groot Koerkamp (eds.) Precision livestock farming '09. Papers presented at the 4th European Conference on Precision Livestock Farming, Wageningen, The Netherlands, 6-8 July 2009.

[4.1] WASP project, Deliverable D4.3 ”Parameters, protocols and benchmarks”, August 2007.

[4.2] WASP project, Deliverable D6.2 ” Specified application requirements”, September 2007

[4.3] http://www.omnetpp.org/

[4.4] http://wiki.github.com/mobility-fw/mf-opp4

[4.5] W.M.P. van der Aalst, M. Voorhoeve, "Simulation Handbook", Computing Science Reports 95/32, Eindhoven University of Technology, Eindhoven, 1995, http://wwwis.win.tue.nl/~wvdaalst/publications/p30eng.pdf
[4.6] IEEE std. 802.15.4 - 2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)
http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf
[4.7] Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks with IEEE 802.15.4: Jose A. Gutierrez, Edgar Callaway, and Raymond Barrett
[4.8] M. Ahmed, C. Saraydar, T. ElBatt, J. Yin, T. Talty, and M. Ames, “Intra-vehicular wireless networks,” in Proc. of the IEEE Globecom Workshops, November 2007, pp. 1–9.
[4.9] T. ElBatt, C. Saraydar, M. Ames, and T. Talty. Potential for intravehicle wireless automotive sensor networks. In 2006 IEEE Sarnoff Symposium, 2006.
[4.10] Hsin-Mu Tsai, W. Viriyasitavat, Ozan Tonguz, Cem Saraydar, Timothy Talty, Andrew MacDonald, “Feasibility of In-car Wireless Sensor Networks: A Statistical Evaluation,” IEEE SECON 2007.
[4.11] Hsin-Mu Tsai, Cem Saraydar, Timothy Talty, Michael Ames, Andrew MacDonald, Ozan Tonguz, “ZigBee-based Intra-vehicle Wireless Sensor Network,” IEEE ICC 2007 WAS.