A communication framework for Pedestrian Detection System

A dissertation submitted to the
University of Dublin, Trinity College,
in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science.

Maciej Wieckowski

May 2006

DECLARATION

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

_____

Maciej Wieckowski,

29 May 2006

PERMISSION TO LEND AND/OR COPY

I, the undersigned, agree that the Trinity College Library may lend and/or copy this dissertation upon request.

_____

Maciej Wieckowski,

29 May 2006

ACKNOWLEDGEMENTS

I would like to thank Professor Vinny Cahill, my supervisor for his help, support, and technical expertise.

I would also like to thank Marcin Karpinski, for his help during all stages of the development of the system as well as for a helping hand during the tests and evaluation of the system.

Special thanks to Przemyslaw Madej for sacrificing his weekend in order to help me with indoor system testing.

I would like to take this opportunity to thank Matthew Fay for proofreading this dissertation and all my UBC classmates for their friendship and making these two years more enjoyable.

Finally, I would like to thank my parents for great support throughout the demanding time I spent in Trinity College and Vera Goncalves for motivating me throughout the last year.

ABSTRACT

As the number of vehicles on roads is increasing, so is the danger of accidents. Between the years 1992 and 2002 approximately 70% of all accidents in Ireland happened on rural roads, which are defined by the National Roads Authority as roads with the speed limit exceeding 40 mph (64.4 km/h),[1]. In this dissertation we describe the design of a communication framework for a Pedestrian Detection System (PDS) that has the goal of decreasing the risk of accidents involving pedestrians on rural roads. The PDS aims to find the optimal way of detecting pedestrians, especially in dangerous places and in bad weather conditions, thus increasing the safety on the road. The system is a part of the bigger Smart Roads project carried out by Trinity College Computer Science Department.

The solution we describe is based on the beacon approach and makes an assumption that pedestrians carry beacons that may for example be embedded in high visibility vests. The beacons send radio signals periodically which are received by Mica2 motes placed along the road. The strength of the radio signal (RSSI) received by the motes placed on the road is then sent to a base station for further processing. The base station infers the location of the beacon in relation to the motes placed along the road using acquired information and multicasts the location information over an Ethernet connection using User Datagram Protocol (UDP). The framework makes the assumption that the data will be send over the wireless link, received by a car's on board system and communicated to the driver. The proposed technology considers the use of the 802.11b protocol used in combination with directional antennas which should allow for establishing the connection over a range of 400-500 metres. In this paper we present the software developed for both the Mica2 motes written in NesC as well as the software for the base station written in Java. We also describe the routing protocol designed for the motes that allows them to communicate the data to the base station over the network of interconnected motes.

We present the results of tests that evaluate the suitability of the hardware used in the project. The tests of the software measure it's performance and investigate whether it may be treated as a starting point for further development.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: Introduction

1.1 Motivation

The project is motivated by large number of accidents that take place every day, especially on rural roads. Many fatalities caused by these accidents involve pedestrians. Over 20% of all people loosing their lives in car accidents are pedestrians.

Even though the number of fatal accidents seem to be gradually decreasing after the year 2000, it is very important to try to provide systems that will make it easier for the driver to notice potentially dangerous situations and help him/her to react faster thus reducing the risk of an accident. It is worth mentioning that even though the number of the car users that were killed in road accidents started to decrease, the number of pedestrians that are killed on Irish roads is growing. The system aims to decrease the overall number of accidents, but focuses mainly on safety of the pedestrians.

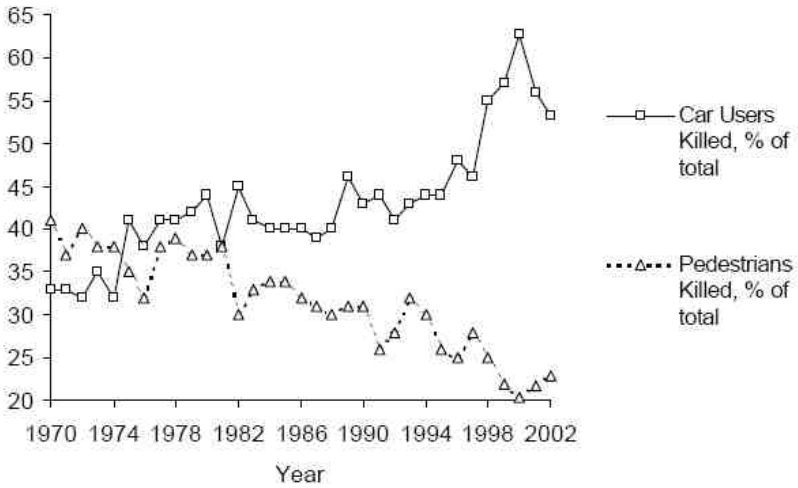

Fig 1 Percentage of car users and pedestrians killed on roads[1]

Rapid development in mobile, ubiquitous and wireless technologies is starting to encourage researchers to find better solutions for transportation. New technologies

allow us to try to enhance existing vehicles, especially cars, to provide maximum safety and comfort.

With the recent appearance of the hardware for wireless sensor networks it gives new opportunities concerning the situation on the road. A wireless network of sensors that runs context aware, real time applications is a potential solution to part of the safety and traffic problems. It may not only inform the driver about the traffic on the road, but may also participate in the process of generating warning messages that may concern bad weather conditions or collisions on the road. The scope of different applications is vast and will surely be explored in the nearest future.

On the other hand, the evolution of wireless short and long range technologies makes it feasible to communicate this information from the systems on the road to the vehicle that is moving at high speeds. These technologies provide low latency and high throughput, while guaranteeing the delivery of the data. Wireless standards may not only to allow for the communication within wireless sensor network, but also between the sensor network and cars moving within the range of a monitored area. What is more, with more and more vehicles with Global Positioning Systems and wireless enabled devices installed, it is possible to make this process even more successful by enabling inter car communication. Ad-hoc networks of vehicles would be able to exchange relevant information and warn each other about potential safety issues.

All these technologies motivate us to look for the ideal solution that may increase safety of all roads users in Ireland.

1.2 The definition of an Intelligent Transportation System

Intelligent Transportation Systems (ITS) take advantage of a wide range of different wireless and wired communication technologies which are integrated in the transportation infrastructure, as well as vehicles, and provide mechanisms which relieve congestion and increase safety.

The area of ITS is vast and includes many different solutions. Many researchers focus their attention either on a different part of ITS or simply design different solutions that aim to solve different problems. When we are thinking about ITS, we should not consider it only as a hardware or software solution. We should think about

it as a complete set of different software, hardware and communication technologies that are used in order to provide better understanding of traffic patterns, route planning, safety on the road and many other aspects of transportation.

Intelligent transportation systems does not only aim to provide the information to the driver. They cover many more issues concerning infrastructure, traffic management and much more.

U.S. Department of Transportation which is carrying out an extensive research project on Intelligent Transportation System makes the following division of ITS:

- Intelligent Infrastructure
    - Arterial Management
    - Freeway Management
    - Transit Management
    - Incident Management
    - Emergency Management
    - Electronic Payment
    - Traveller Information
    - Information Management
    - Crash Prevention and Safety
    - Roadway Operations and Management
    - Road Weather Management
    - Commercial Vehicle Operations
    - Intermodal Freight
- Intelligent Vehicles
    - Collision Avoidance Systems
    - Collision Notification Systems
    - Driver Assistance Systems

This is only one interpretation and characterisation of an Intelligent Transportation System. However, it gives a sound overview of what the areas of interest are in ITS.

There are many applications which try to use different technologies and have different approaches. Most of them fit within the boundaries of the structure described above.

This dissertation aims to explore both the intelligent transportation infrastructure and intelligent vehicles which are a focus of interest for many researchers and car manufacturing companies. Intelligent vehicles are the core part of Intelligent Transportation Systems (ITS). They use different sensing techniques and control algorithms to assist the driver with safe driving. These algorithms may either control the functionality of a car (i.e. Citroen has introduced driving assistance helping to keep the vehicle on it's lane) or provide collision warning systems that may inform the driver of a potentially dangerous situation.

There are also other features of ITS that aim to provide more information about the traffic, location information, help in finding an optimal route in high density urban areas with heavy traffic. These systems are however beyond the scope of this paper and won't be discussed. Intelligent transportation systems include both the inter vehicle communication and road to car communication.

1.3 Objectives

The main objective of this dissertation is to implement the basic communication framework for the Pedestrian Detection System, which may be used as a starting point for the future development.

The goals include:

- The overall design of the framework.
- The implementation of a routing protocol for ad-hoc network, allowing for successful detection of pedestrians and providing gathered information to the higher levels of the system for further analysis.
- The implementation of an application for the part of the system, that will be capable of gathering the information, inferring the location of a particular pedestrians and broadcasting the location information over a wireless link.
- The implementation of a basic client software for the system that may be installed on the car's system (further referred as a car's on board system), and

be able to receive location information from road infrastructure of the system over a wireless link.

- The introduction of security measures within the implemented design to provide resilience from malicious users.
- The evaluation of accuracy of the system, susceptibility to latency as well as scalability.

1.4 Document Structure

This is a brief description of the document, giving the overall layout of the dissertation and a short description of issues addressed in particular chapters.

1. **Introduction**

   Gives a brief overview of the subject and motivates the development of the system. It also defines the idea of Intelligent Transportation Systems (ITS) and the Pedestrian Detection System (PDS) as a variation of ITS.

2. **State of Art**

   Describes various technologies and solutions that are either used in systems that have similar functionality to PDS or fulfil harsh requirements imposed by a demanding real time system.

3. **Design**

   Gives an overview of the communication framework and location strategy used in PDS as well as describes algorithms designed for this solution.

4. **Implementation**

   Gives implementation details of the system, including hardware used in the project, the software designed in two different programming languages and the types of communication procedures within the system

5. **Evaluation**

   Discusses the results of the tests carried out in order to identify the accuracy and the performance of the system as well as presents conclusions concerning further development.

6. Conclusions

Concludes the paper giving an objective opinion on the present implementation of the system, outlines possible improvements and discusses future work on the communication framework.

Chapter 2: State of Art

2.1 Related work – Vehicle based pedestrian detection

Most of the Pedestrian Detection Systems are not a part of the infrastructure of an Intelligent Transportation System. The vast majority of projects carried out within this area are standalone applications implemented as one of the features of Intelligent Vehicles and are mostly based on computer vision and the analysis of the images acquired from a digital camera. This approach usually involves a monocular camera installed in the car and observing the road in front of the vehicle. Some systems that are already implemented are based on fairly simple mechanisms, others aim to be more sophisticated and allow to track pedestrians in various environments and conditions, using complicated algorithms to distinguish pedestrians from the background and other moving objects. This methods include shape-based methods, texture and template-based methods, stereo, as well as motion clues and neural nets-based methods.

The system described in [2] is a computer vision system based on images recorded by a digital camera, that is used to recognize pedestrians in different environments and localizing them using Kalman filter estimator. The system consists of multiple components that perform different tasks:

- "Pre-attentive Phase" – which is a low level vision elaboration.
- "Symmetry Detection" – is a process that evaluates the symmetry maps.
- "Bounding Boxes Generation" – creates pedestrians outlining.
- "Bounding Boxes Filtering" – selects pedestrian boxes.

- “Pedestrian Localization” – localises pedestrians using the spatial position of the boxes.

- “Bounding Boxes Tracking” – evaluates the accuracy of the estimation.

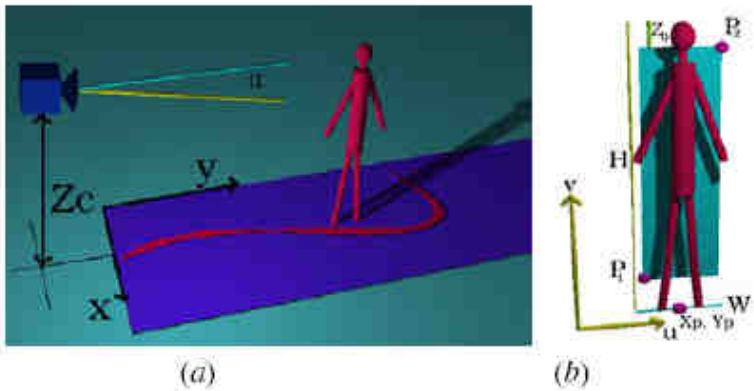

Fig 2 Setup scheme and problem variables: (a) world coordinate reference system, (b) image coordinates of the scene bounding box.

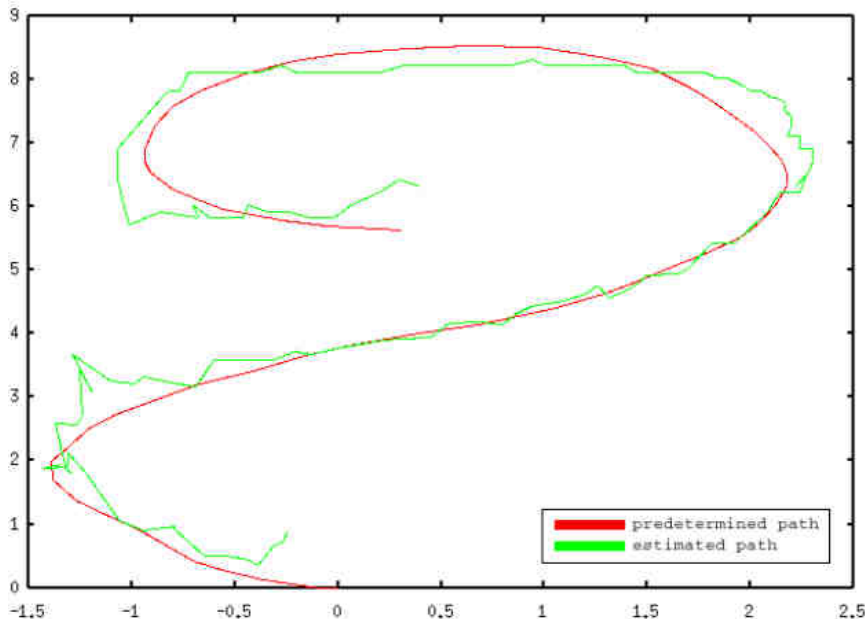

Fig 3 Predefined and estimated path of a slowly walking pedestrian

According to experiments, the creators of the system have noticed, that the overall performance of the system is good. The system was accurate in spatial localisation of a walking pedestrian in indoor environment and performed equally well in outdoor settings.

The system described above used daylight images and thus would not be useful in bad weather conditions or at night.

Another systems that have already been developed, relies on infrared images. These systems seem to be more useful in low visibility scenarios.

The system described in [3] works on low quality infrared video images. The design uses the idea of probabilistic templates to find the shapes of pedestrians in infrared images which are very often distorted. The authors outline that infrared images provide relatively small area of the image that is of any interest, which reduces the probabilistic template search and thus helps to identify people on the road. They outline many problems related to infrared images and describe the system as a solution complimentary to another system already installed in the vehicle.

In general, existing pedestrian detection systems are far from being accurate and show high dependency on environmental conditions. Computer vision is not accurate and suffers from similar constraints and problems as a human eye. This approach may however turn out successful thanks to the evolution of computer vision systems and new methods of analyzing digital images.

2.2 Infrastructure based pedestrian detection - Intra system communication

The communication within the internal parts of the infrastructure based pedestrian detection system is in the area of ad-hoc networking and unstructured inter node communication and may be based on existing ad-hoc routing protocols as well as a dedicated implementation. To design a reliable and scalable solution we need to examine existing routing protocols and identify the features that would be relevant to the Pedestrian Detection System being designed in Trinity College.

2.2.1 Ad-hoc networks

Wireless networking emerged in 1970s and since that time was gradually gaining popularity. However, more rapid growth started in the 1990s, mostly thanks to the 802.11 standards.

At the moment we can distinguish two different types of wireless networks – infrastructure networks and ad-hoc networks. In this dissertation we will focus our attention on ad-hoc networking. This type of networks has no structure at all. There are no fixed routers, many nodes in ad-hoc network act as routers at the same time.

There is a whole range of different routing protocols available for ad-hoc mesh networks. We can divide them into two main groups:

- Proactive
- Reactive

2.2.1.1 Proactive ad-hoc routing protocols

Proactive routing protocols try to maintain up-to-date information about the nodes that are connected to the network. They require each node to maintain routing tables containing the addresses of the neighbours. They differ when it comes to calculating the metric, the way the information about the network are gathered and the routing process itself.

One of the examples of proactive protocols is Destination-Sequenced Distance-Vector routing protocol (DSDV). It is based on classical Bellman-Ford routing mechanisms. This approach states that every node needs to maintain a table of all destinations, together with the number of hops to each destination. Each entry in the table is associated with a sequence number assigned by the destination. It is used to distinguish between up to date and stale entries. In order to maintain a routing table consistency, routing table updates are sent periodically. These periodical updates can generate a lot of traffic. To reduce the amount of bandwidth within the network, two types of updates have been introduced. One of them is “full dump” and contains all routing information. This may require to use multiple PDUs. Another type is “incremental”. It only contains information that changed since the last update and usually uses much less bandwidth. The routes include the address of the destination, number of hops, a sequence number unique for the broadcast and a sequence number of the destination. [4]

2.2.1.2 Reactive ad-hoc routing protocols

Reactive routing protocols, known also as on-demand routing protocols create much less overhead. Instead of keeping large routing tables of many motes that may never be contacted, they initiate a route discovery of a destination node when there is a need of contacting it. When a route is found, it is kept in the routing table until it expires or the destination becomes unreachable.

Ad Hoc On-Demand Distance Vector Routing (AODV) is among the two most popular reactive routing protocols used currently. It provides different types of communication: broadcast, multicast and unicast is available. Route discovery in AODV is initiated only when a source node requires to contact the destination or join a multicast group. The maintenance of the routes is limited. Entries are kept in the routing table as long they as they are used or until the multicast group exists. AODV provides mechanisms for avoiding routing loops (sequence numbers).

If a route is needed, a route request (RREQ) is sent to the network. Either the nodes that have the route to the destination or the destination itself can send an answer to this request. When the destination is found, it responds with a RREP packet.

In case a link between the two adjacent nodes is broken, a RERR packet is sent, which informs abut an invalid link. RERR is propagated to the source to inform that the destination is no longer reachable.

Another advantage of AODV is the fact that it additionally supports multicast.

The Dynamic Source Routing (DSR) as a competitive solution to AODV. It also is a reactive routing protocol. It is based on a totally different concept. Instead of keeping routing tables, nodes on the network maintain route cache and the packets are source routed. Entries in the route cache are updated as the node learns new destinations. There are two main phases in DSR:

- Route discovery
- Route maintenance

If a node wants to route the packet to the destination, it first checks it’s route cache. If it finds a valid route to the destination it source routes the packets (the packet header contains the whole route to the destination). However, if there is no route to the required destination, the node broadcasts a route request packet. The packet

contains the address of the destination, source node address, and an ID number. The packet is then propagated through the network. The receiving node checks it's routing cache for the destination address. If it does not have a valid entry for the destination, it adds it's own address and forwards the packet. To limit the number of route request packets, every node receiving this packet checks if it wasn't already forwarded through this node, by looking at the route path. If it sees it's own address in the path, it does not forward the message.

If the route request packet reaches the destination or the node that has a valid route to the destination in it's route cache, a route reply is created. Depending on whether symmetric links are supported or not, a destination or a node that knows the route to the destination may either reverse path to the initiator of a request message or send it's own route request packet to find an appropriate route to the initiator.

The route error messages and acknowledgments are used for route maintenance. Error messages are generated when a link error is encountered. It is generated to inform other nodes about a stale link. On the other hand, the acknowledgments are used in order to verify the operation of route links. [4]

2.3 Infrastructure based pedestrian detection - Inter system communication

There are a few possibilities concerning the communication between the road infrastructure and the vehicles as well as inter car communication. They are a both a subject of similar constraints and limitations. The problems that will occur cover such issues like the short wireless range of some technologies and the main concern, which is the very short period of time within which the communication will have to be established and completed. These limitations however concern only short range technologies. The use of long range technologies like 3G would eliminate those issues.

2.3.1 Short range technologies

In the project we considered various wireless technologies. One group of communication standards concerned short range radio communication. This was

motivated by the fact that, exchanging the information over a short range wireless link, does not involve the necessity of using the services of a third party thus reducing at least the initial costs of development and increasing the ease of deployment and implementation of the framework.

2.3.1.1 Dedicated Short Range Communications (DSRC)

Dedicated Short-Range Communications (DSRC) is a technology that is built on top of 802.11a standard and uses the spectrum of 5.9GHz to transfer the data over a wireless link. It is perceived as an emerging standard for Road to Vehicle Communication (RVC) and Inter-Vehicle Communication (IVC) systems. The 5.9Ghz band consists of seven channels (ten megahertz). One of them is a control channel, the other six are service channels. DSRC involves both vehicle-to-vehicle and vehicle-to-infrastructure communications and is expected to support both safety/public safety and non-safety applications. The focus is however on safety applications as the non-public safety use of the 5.9 GHz band is perceived as inappropriate if it leads to degrading the performance of safety/public safety applications.

Classification		DSRC-type RVC	
ARBID STD		T55	T75
Service		ETC	ETC Information Shower
Specifications	Frequency band	5.8 GHz band	5.8 GHz band
	Frequency channels	4 Ch	14 Ch
	Dedicated bandwidth	8 MHz	4.4 MHz
	Transmission rate	1 Mbit/s	1.1/4 Mbit/s
	Wireless access method	TDMA-FDD	
	Modulation method	ASK	ASK/QPSK
	Service area (reference values)	Up to 30m	Up to 30m
Features	Mobility	Consideration for the reduction of time during link establishment	
	Other system reference	No interference	
	System capacity	Bu number of slots (maximum eight slots per channel)	

Fig 4 Road to Vehicle DSRC

The main attribute of safety applications is that fact that they aim to save lives by warning drivers about potentially dangerous situations and give them additional time to react. Therefore, the availability, reliability and low latency are the basic requirements of such applications. [5]

2.3.1.2 WiFi and ad-hoc networks

As wireless enabled handhelds are becoming more popular and are becoming widely used together with GPS modules in order to provide relevant information to the driver, we cannot ignore the possibility to use them as a potential “On-Board Units” that may be used either for inter-vehicle or road-to-vehicle communication. The possible use of external antennas in the cars is the subject of possible feasibility study. These technology, together with ad-hoc routing protocols may be a possible solution of the problems of “Smart Roads Project”.

802.11b standard operates in 2.4GHz ISM spectrum. The original 802.11 wireless standard specifies the maximum throughput of 1 Mbps and 2 Mbps. As a medium it uses radio waves and a technique called frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS). They are both two different signalling mechanisms. They are not interoperable.

The frequency hopping technique divides the 2.4 GHz band into 75 1-MHz sub channels. Both the sender and a receiver agrees on a hopping pattern, and data is sent over a sequence of the sub channels. Each exchange of the information within the 802.11 network occurs over a different hopping pattern. These patterns are designed to minimize the chance of two senders using the same sub channel simultaneously and reducing the risk of a collision. [6]

The data link layer consists of two parts- Logical Link Control (LLC) and Media Access Control (MAC). It follows IEEE standards and uses 48bit addresses for MAC. 802.11 uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as opposed to regular Ethernet (Carrier Sense Multiple Access with Collision Detection (CSMA/CD)) .

Currently most widespread version of 802.11 standard is b/g version. It provides high speed network access with the bit rates of 11/54Mbps. 802.11g provides higher data rates as well as stronger security (WPA encryption).

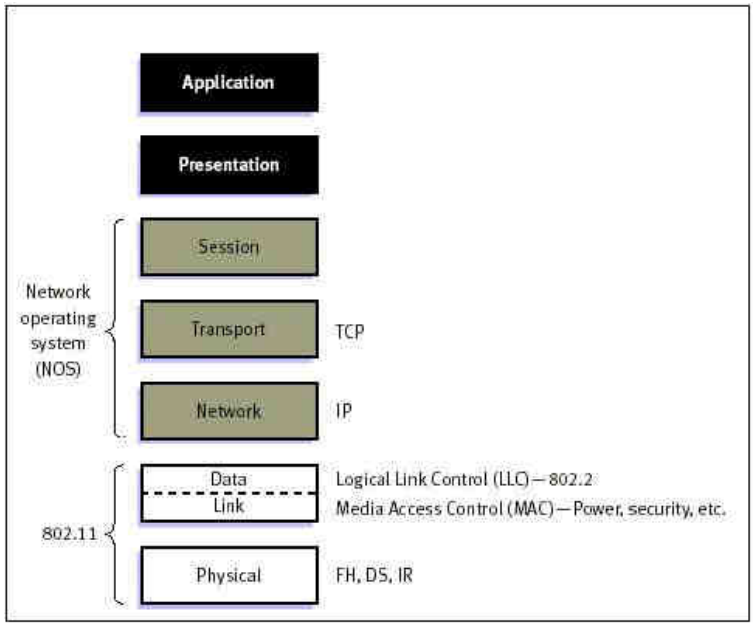

Fig 5 802.11 and ISO model

2.3.1.3 Parasitic routing

Parasitic Routing is a technology that is in the early phase of research and development. The idea of store-and-forward algorithm seems a good solution for this project as well. It enables the cars to store important information for specified periods of time and forward them as soon as the destination becomes available.

“The major difference between the store-carry-and-forward paradigm and the store-and-forward paradigm employed in the majority of routing protocols is that the former of the message takes advantage of additional resource of a node, its mobility. Store-and-forward protocols take advantage of an agent’s buffer, transmission capabilities, some processing time, and some energy, while store-carry-and-forward protocols use all the previous, and use a nodes mobility to enhance the likelihood of delivery, in doing so a message will most likely be held in an agent’s buffer, or queue, for an extended period of time also. The mobility may be random, designed in advance to deliver messages, or may be dictated on-the-fly in an attempt to ensure speedy delivery of all messages.”[7]

The idea of storing the messages and carrying them around until their TTL (Time To Live) expires makes a lot of sense, especially on rural roads, where traffic density is not high. Even on the motorway it might not be possible to establish and sustain communication with adjacent vehicles at all times. In that case, buffering data packets may greatly increase the delivery rate of messages. This applies mostly to short range technologies like 802.11b/g.

2.3.2 Long range communication

The use of long range technologies like 3G eliminates most of the problems that emerge while trying to communicate to nodes moving at high speeds. It may however imply the need of involving third parties like mobile operators in routing architecture. This approach may not only increase the costs of implementing the system, but in particular situations might also increase the latency in the delivery of crucial information.

2.3.2.1 3G and GPRS

Both UMTS and GPRS are currently widely available. Both of the technologies allow for acceptable data transfers and due to high mobile network coverage may give satisfactory results in road-to-vehicle communication.

3G is mobile providers' technology. The services are provided by companies operating their own wireless networks and offer their services to end-users. The advantage of 3G is that a mobile base station can provide service to nodes that are as far as several kilometres away and moving at high speeds (up to 100km/h). The base stations are interconnected by backhaul network and provide access to standard PSTN network. Currently 3G offers the data throughput of 384kbps. In the future the specs of the standard take into consideration expanding the available bit rate to 2Mbps. [8]

GPRS which is a an older standard than 3G (often referred to as 2.5G) is the packet-oriented extension of GSM. It relies on the re-use of the radio infrastructure of GSM while introducing new network nodes in the core network providing the required packet switching functionality. GPRS is mostly intended to provide better service for

Internet applications in comparison to existing circuit switched services that are provided by GSM. It allows data transmission exceeding 100kbps. GPRS is packet-oriented, which makes it possible for many users to share the scarce radio resource. Thanks to this approach flexible access is possible while idle users can still be online.[9]

2.3.2.2 Interoperability for Microwave Access (WiMax)

WiMAX stands for Worldwide Interoperability for Microwave Access. It describes the systems that pass interoperability tests for the IEEE 802.16 standards.

Products that pass the conformity tests for WiMAX are capable of forming wireless connections between them to permit the carrying of internet packet data. The idea of WiMax is in some ways similar to WiFi. It however, has many improvements that allow to carry the signal at high bit rates over longer distances than in WiFi.

The physical layer specifies the frequency range between 10 and 66GHz for line-of-sight configurations. It uses single carrier modulation. Both the TDM and FDD technology are used on uplink and on downlink. The standard 802.16a specifies the range of frequencies between 2 and 11GHz and allows for establishing non-line-of-sight connections.

MAC layer is designed to support various protocols and services (TDM, IP, VoIP, ATM, GFR). It also provides bandwidth allocation and QoS mechanisms, including unstandardized scheduling and reservation management. Authentication is provided by privacy sub-layer.

There are three MAC Sub-layers:

- Service-specific Convergence Sub-layer
- Common Part Sub-layer
- Privacy Sub-layer

802.16 supports point-to-multipoint architecture A base station is a central point that handles multiple independent sectors. Downlink covers the transfer of the data to service subscriber and is multiplexed in TDM fashion. Uplink on the other hand is shared between service subscribers in TDMA fashion,[10].

2.3.3 Hybrid approach to ITS communication framework

In the project that needs to fulfil very strict constraints, due to accuracy requirements greatly influencing safety on the road, the latency of delivering the information must be low. What is more, the rate of success in delivering these messages needs to be very high. To provide such high quality of service, it might be necessary to incorporate many long and short range technologies to be sure, that the driver will be informed about the current situation on the road as soon and as accurately as possible.

2.3.3.1 FleetNet Project

The FleetNet Project aims to develop a framework for inter-vehicle communication. It's purpose is to locally distribute data between the cars in order to increase safety of derivers and passengers. It takes into consideration location awareness of the application. It also integrates the Internet with the system, so the project is not only a solution working locally, but it brings two concepts together – the Intelligent Transportation Systems and the Internet.

FleetNet aims to use existing radio hardware. The communications protocols developed are supposed to be implemented on top of the hardware system selected.

The assumptions are that the system must be able to transmit information at the rate of at least 1 Mbps, at the communication range of at least 500 m.

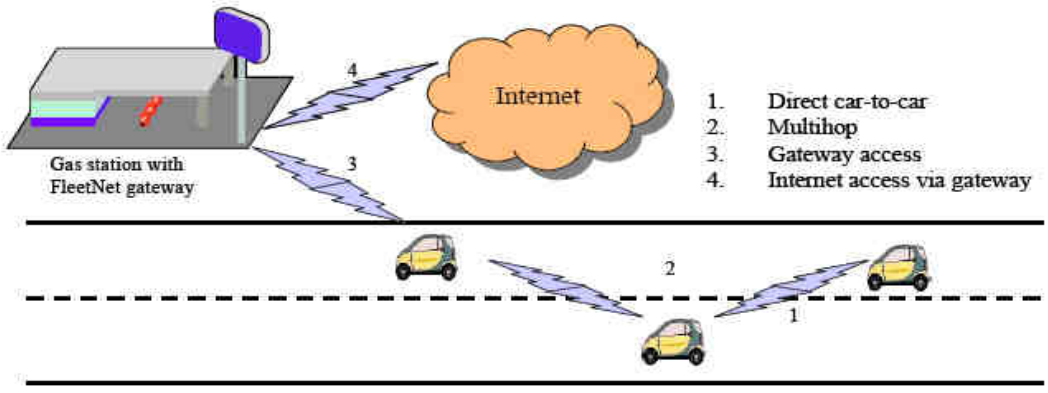

Fig 6 A simple FleetNet scenario [11]

There are three different classes of applications in the FleetNet project:

- Cooperative Driver Assistance Applications – this type of applications is supposed to distribute sensor and other status data among vehicles. The use of this applications is going to be limited mostly to safety notifications like accidents that happened on the road or emergency braking.
- Decentralized Floating Car Data Services (dFCD) – using the concept of location awareness
- User Communication and Information Services (Internet access) [11]

The basis for radio communication is ULTRA-TDD which has also been implemented in 3G mobile phones. “It offers high flexibility with respect to asymmetric data flows, allows communication over great distances, and supports high speeds. Another essential argument for the use of UTRA-TDD as a basis for FleetNet is the availability of an exclusive unlicensed frequency band in Europe (2010–2020 MHz), which offers two completely separate operating channels. Furthermore, the system allows high granularity for data transmission due to its CDMA component. However, because of the original cellular design of UTRA-TDD, a new ad hoc mode will have to be developed within the FleetNet project. The UTRA-TDD radio hardware provides a communications range of approximately 1 km and bit rates up to 384 kbps and 2 Mbps, depending on the absolute and relative speeds of communicating cars.”[12]

One of the major advantages of an ad-hoc network in FleetNet project is that thanks to multi-hop and vehicle to vehicle communication it will be possible to propagate the safety sensitive information to many cars on the road. The benefits of this fact are obvious. Drivers can be warned about dangerous situation on the road much earlier and will be given additional time to react.

One of the examples of an application greatly increasing safety is an accident warning application. In case of a collision, sensors may detect an airbag ignition. This may trigger a process of sending emergency notification to nearby cars informing the drivers about the accident. The messages may be forwarded to other cars, much further on and notify the drivers of an incident and recommend, for example, that they should slow down and be extremely cautious.

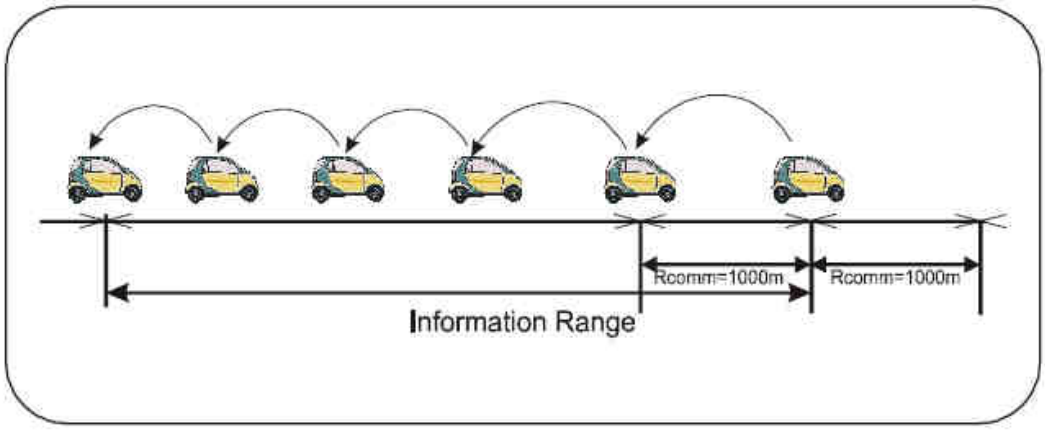

Fig 7 Transmission Range vs. Information Range [12]

The algorithm proposed in [12] allows for sending the messages through multiple hops. It limits down the range within which the information are propagated, and prevents the messages to be forwarded infinitely.

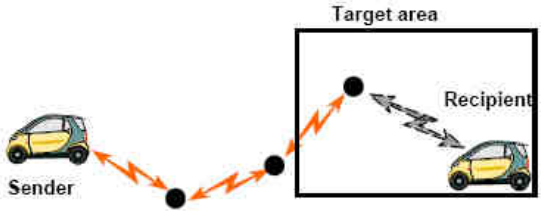

Fig 8 Geographic addressing and routing [13]

As mentioned before, the applications in FleetNet are categorized in three different classes. In the protocol architecture, the application types are mapped to two different communication classes:

- FleetNet aware
- FleetNet non-aware

The major difference between these classes is that FleetNet non-aware applications are simply based on regular IPs, while the applications within the FleetNet aware class have knowledge about the recipients' location. The difference is in the addressing scheme. The aware applications define a two dimensional space which limits the area where the recipients are located. The examples of these kind of applications cover the emergency notifications and simplify information about the traffic.

Routing the information over multiple hops and using the information about the location of particular nodes is called geographic routing. Geographic routing messages can be divided into two groups- messages sent to single recipients only (unicast) or a group of recipients within a certain area (geocast).

The creators of geocast in FleetNet proposed a solution that consists of two different steps. The first one is called “line forwarding” and simply forwards the packet to the area where the recipients of the message are residing. The second one involves forwarding the message to all the nodes within the area and is called “area forwarding”. The idea of two stage geocast is shown in Fig 7.

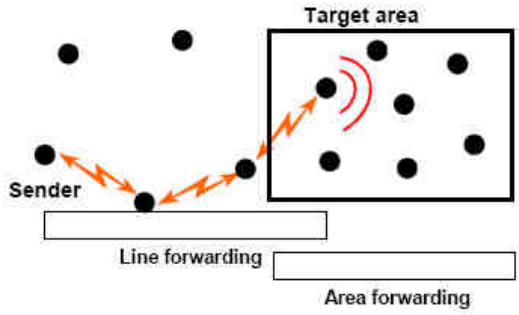

Fig 9 Line and area forwarding [13]

The first forwarding step can be carried out using for example a greedy routing scheme. It assumes that the nodes know the location of their neighbours and the forwarding node simply sends the data to the node or nodes that are geographically closer to the destination region. This process is repeated until the message arrives at the node inside the destination area. At this time line forwarding is stopped.

When a node inside a desired geographical area is reached, the area forwarding can be carried out. If a receiving node knows the route to the destination, then it can immediately forward the packet. This however very often is not the case. If a receiving node does not know the route to the destination, it sends a route request which is similar to the approach known from traditional ad-hoc networks. After receiving the position of a destination node a common routing approach is applied. The packets are then forwarded using a regular line forwarding. As the node receiving the packet from the initiator of the message is much closer to the destination within a region, it will have more up to date information about the destination and thus will be able to infer the destination region that the message should be forwarded to faster than the initiator. [13]

The FleetNet project is a complex solution and it takes advantage of the use of many different technologies and ideas. Most of them are beyond the scope of this paper. We only focused our attention on the solutions that are relevant to the Pedestrian Detection System communication framework.

Chapter 3: Design

3.1 Overview of a "Pedestrian detection system"

The system consists of several parts that have different functionality and carry out different roles in the overall deign.

This chapter looks at possible detection mechanisms allowing for successful location of pedestrian on the road, as well as the analysis of the data and the process of communicating the inferred information to vehicles within the geographic area of interest. The initial concept involves the use of forwarding motes placed along the road, capable of sensing the signal broadcasted by a beacon carried by a pedestrian and measuring the strength of the signal received from a broadcasting node.

3.2 Sensing the presence of a pedestrian

We assume that sending the data directly from the motes to the car may not be possible. The maximum range of some of the motes modules are as little as 75 metres. Assuming that the vehicle may be moving at the speed of 100km/h (it moves by 27.77 metres every second), even if we assume perfect conditions for radio wave propagation and that the motes use their full communication capabilities, it is still very unlikely that it will be feasible to establish and send all necessary information within less than 3 seconds. Instead the information might be kept and updated on the gateway and send over to the car using long range communication standard.

The most important issue when implementing the system is latency. It should be reduced to minimum so that the information delivered to the gateway is as up to date as possible. However, the time constraints for this process are not extremely high. As pedestrians moving along the road don't travel at high speeds and the accuracy of the position of the pedestrian does not have to be extremely high (5 – 6 metre accuracy is

enough), the data stored on the gateway should be updated approximately every 2 seconds.

3.3 Analysis of the process of detecting a pedestrian (beacon approach).

For the purpose of this project the assumption has been made that the system will use beacons in order to sense the presence of a pedestrian on the road. The process of inferring the location of a person involves the use of measurements of the quality of the signal, which is sent from beacons carried by pedestrians and received by one or more motes on the road. This simple approach involves inferring the position of a person based on the fact that the beacon is visible to the motes within a certain range. The range of various radio modules alternates between approximately 30 and 300 metres line of sight. The range can decrease due to bad weather conditions, obstacles between motes, and the shape of the terrain. Placing the motes on the road every 5 metres, should give sufficient resolution of the system and provide the information accurate enough to successively calculate the location of a pedestrian and use this information to generate a message that will be communicated to the driver. The layout of the motes on the road and the distances between the motes will be tested and evaluated. The final decision concerning the layout of the motes forwarding the information will be based on experiments in the environment as similar to real life conditions as possible. The real range of the radio modules of the forwarding motes and the beacons will determine the distance between the motes so that the presence of the beacon can be inferred.

In order to decrease the latency of the network, the information about the pedestrians on the road may be stored at the gateway, which will be updated in real time about the movements of those pedestrians. The gateway will be waiting for a vehicle to come in range and then will send the data.

The information will be routed to the base station using motes along the road. Present approach involves the use of the routing protocol designed for the Pedestrian Detection System. It is based on broadcasts and does not use individual nodes' addresses to forward the information.

To conserve the battery life, some sort of reasoning concerning the movement of the beacon can be implemented in the next stage of the development. The person that walks along the road, usually moves with the speed of 5 – 6 km/h. If a beacon is monitored for a certain amount of time and the direction of the moving beacon is reasoned according to the sequence of readings, the intervals between the moments when each beacon is monitored can be lengthened. It is very unlikely that the person walking along the road for a couple of minutes will unexpectedly start running in the opposite direction. Even if it happened, the change in the interval between which the beacon is controlled by 2 or 3 seconds will not make much difference. It is very important to remember that a car moving with the speed of 100 km/h moves approximately 25 metres every second. A few metres inaccuracy in the location of the beacon is not going to influence the overall performance of the system.

3.4 Initial design and system improvements

The following diagram shows the initial design of the system (Fig 10). During the process of implementing the communication framework, the whole range of different technologies have been eliminated. The major change, has been made in the process of inferring the location of a pedestrian. As you can see from the diagram, initially we treated the forwarding motes as a distributed system that would calculate the location of the beacon in relation to motes receiving the signal and then just send ready data to the base station to be forwarded to vehicles.

It turned out that the centralised approach is not only easier to implement, but it should also allow to conserve battery power as all of the processing is moved to the base station which uses external source of energy while the motes rely only on internal batteries. What is more, the motes only forward the data so the risk of loosing a packet or delay caused by the nodes that are busy with processing the information is much lower.

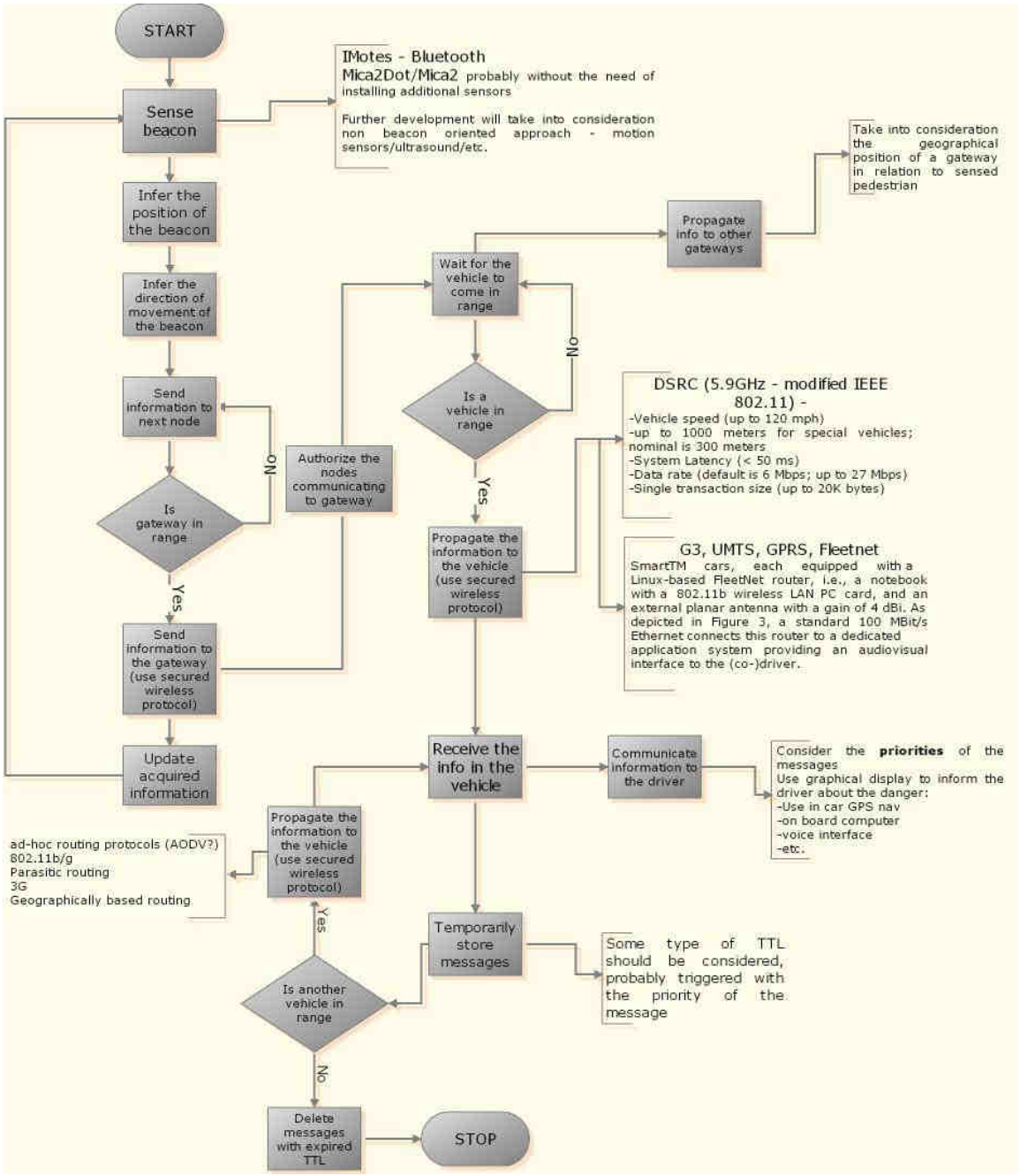

Fig 10 Initial diagram of the system

The following image (Fig 11) depicts the present architecture of the framework. It moves the burden of the analysis of gathered data to the base station thus reducing the processing requirements of the motes in the ad-hoc network.

This approach also makes the process of software development much faster and easier to debug as it reduces the number of possible sources of an error.

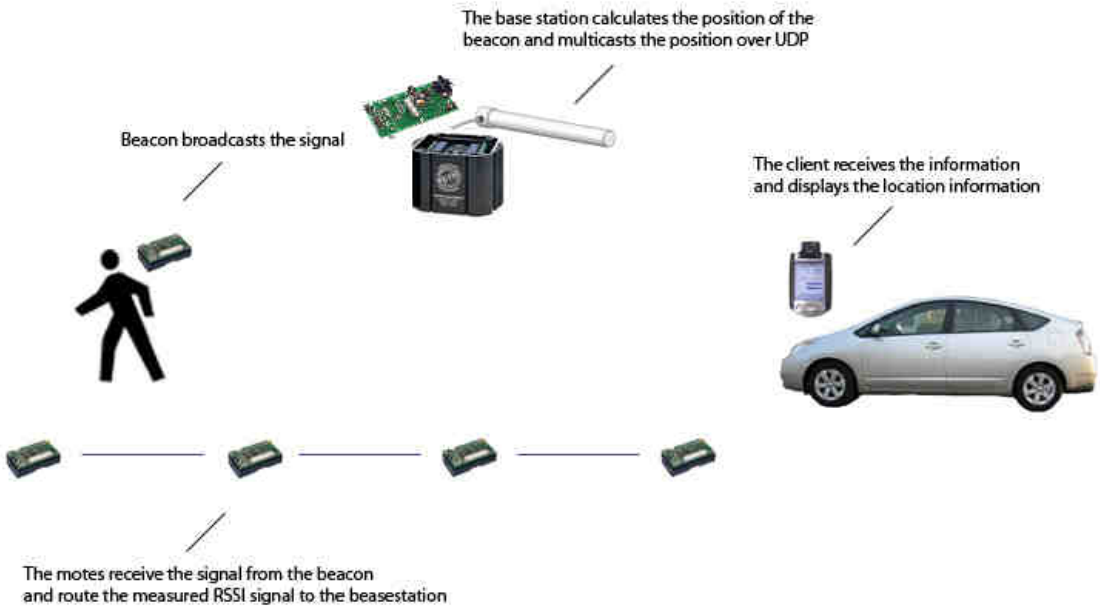

Fig 11 Structure of the communication framework

3.5 Inter mote communication

Most of our attention was focused on inter mote communication and the optimisation of the routing protocol. As a base station is a relatively powerful machine, the process of inferring the location doesn't influence the performance. Most of the delay is created while the data is collected and routed to the base station. This is why we put so much work in the design and implementation of the routing protocol.

The approach we have taken assumes that the motes broadcast every packet they receive, but they never forward the same packet twice. This approach does not create nearly any overhead, because there is no need of keeping any routing tables and checking for active connections with the neighbours. It does require minimal processing. The risk of a broadcast storm is extremely low as the duplicate packets are not re-broadcasted. Each packet also has a time to live field which reduces the life of a packet to 60 hops. This should be enough to forward the data to the base station even from the far end of the segment.

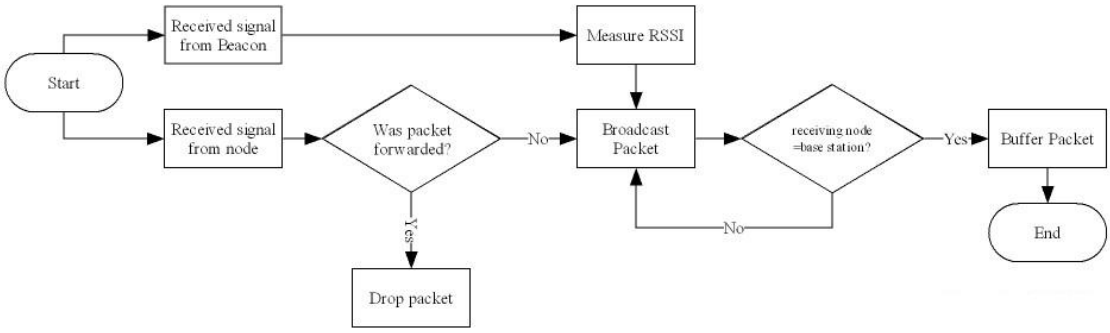

Fig 12 General overview of the routing protocol

One of the obvious advantages of the solution based on broadcasts is that there is no need for organised addressing. The nodes don't need to know what their neighbours are and what addresses they need to forward the information to.

The base station needs to be aware of the location of particular nodes. Present implementation does not include any discovery algorithms. This is the reason why the base station's software needs to know a location of particular nodes on the road.

Until the discovery protocol is implemented it is very important that the nodes in the same segment do not have the same addresses. If they do it will be very hard to distinguish between the packets coming from two different nodes having the same ID. One way to protect the gateway from faulty addressing is to add the addresses of the motes that the packet is forwarded through. This would specify the routing path.

3.6 Mote to gateway communication and the detection algorithm

The detection algorithm is based on the assumption that the packets from beacons are buffered. The base station buffers packets from different beacons, saving the ID of the node closest to the beacon, the timestamp of the packet, and the strength of the signal.

The algorithm iterates through the records of packets with the same beacon ID and compares the strengths of the signal delivered in each of them. It selects the packet with the best signal strength and multicasts the location of that beacon.

The minimal number of packets needed to infer the location of a particular beacon is three. The number of packets used for inferring the location may be much greater, depending on whether the signal broadcasted from the beacon was received by more than one mote on the road.

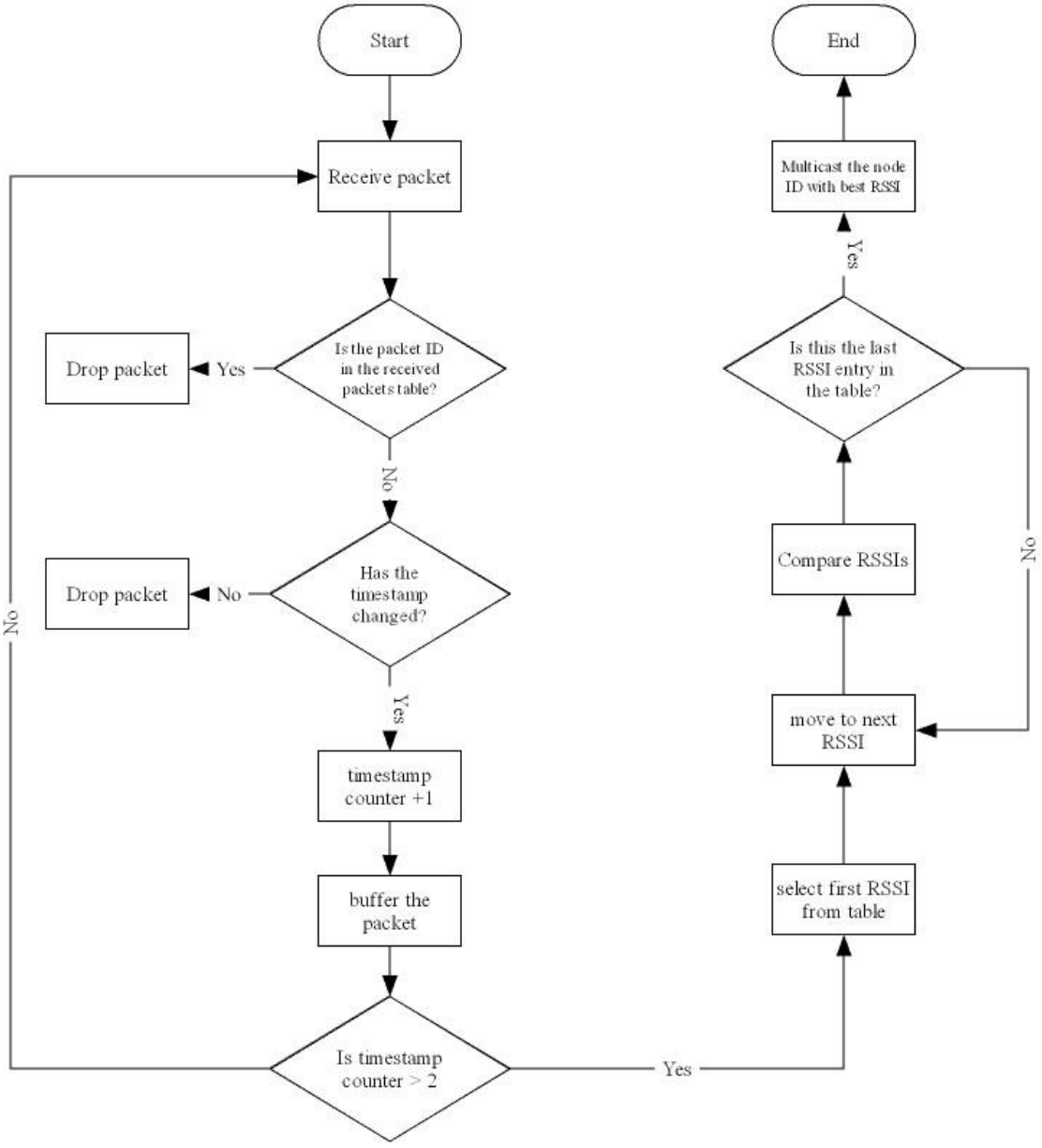

Fig 13 Detection algorithm diagram

3.7 Security considerations

Even though we do not consider any security mechanisms in the initial implementation of the system, there are various issues concerning security that need to be addressed in future development. This is due to the fact that the misuse of this infrastructure may lead to dangerous situations and in result decrease the safety instead of increasing it. As this application is a real time system that has some crucial security requirements, it's very important to provide as much security within the system as is possible in ad-hoc, low resources environments.

3.7.1 The reasons for compromising the system

There are many reasons, which may be the motivation for hacking into the system, bringing it down, or simply feeding false results into it.

One can try to use the system in order to influence the traffic or behaviour of vehicles on the road A malicious user may try to cause an accident or simply stop the vehicle (financial reasons).

Another issue is the privacy of people and cars that will be within the range of the system. Some users might be interested in monitoring the movement of people and vehicles on the road for various purposes.

3.7.2 Major Threats

The misuse of the information gathered by the system may not only pose threats to physical safety of people and vehicles, but also may be the source of the loss of privacy of the data.

3.7.2.1 Privacy

At the present stage of the development we will be using mica2 motes as beacons and all the measurements related to estimating a particular location of a beacon will be associated with the mote ID. Someone who has the information concerning people using beacons with certain IDs assigned would be able to monitor where the person is at a specific moment as well as monitor his/her movements.

Future work will involve different types of beacons, using for example, Bluetooth. In that case, it is important to protect the privacy of the address of the device. It might be dangerous if the Bluetooth address of the device is intercepted. It is not hard to imagine that the intercepted address might be used to get control over a device used as beacon. This is an issue when a device acting as a beacon will be a smartphone or a PDA that is storing for example personal information, account or credit card numbers etc.

One way of protecting such devices is to simply *fuzzyfy* the address. The Bluetooth addresses are 48 bits long and we assume that the system will be used in rural areas with relatively small amount of beacons active within the range of a particular segment of the mote network. This allows for erasing or changing randomly at least half of the addresses not worrying about the problem of distinguishing between different beacons.

The future goal is to use sensors instead of a beacons in the development of the system. In that case the privacy issue concerning pedestrians will rather not be liable anymore.

To provide at least basic level of security, the use of a simple symmetric encryption protocol may be used. This measure would protect the network against data sniffing.

3.7.2.2 Denial of Service

One of the possible ways of bringing down the system would be to flood the network with the information. If it wouldn’t bring down the whole system, it could slow it down, which might become dangerous. The motes have relatively small throughput and processing power, so feeding them with too many requests to route the information to the base station could be block the routing process. DoS attack may also dramatically reduce the life of batteries used by motes and in long term greatly reduce the lifetime of the whole system.

3.7.2.3 Eavesdropping

This seems to be an issue if Bluetooth motes would be used as beacons. We could prevent it by *fuzzyfying* the address. Some sort of encryption algorithm with a pre-

shared key could be used between the motes embedded in the road and the base station.

3.7.2.4 Malicious motes

This includes the use of malicious beacons and forwarding motes. Malicious motes routing the information to the base station seem to be more dangerous for the fault tolerance and keeping the system up and running. Compromising a few motes along the road may stop the routing and bring the whole system down.

Malicious beacons may also feed faulty information to the system. This may result in incorrect number of pedestrians on the road as well as fake position information. The beacons may also be used as nodes performing Denial of Service attack.

Present routing protocol involves the use of broadcasts with sequence numbers, and a mechanism preventing broadcast storms and redistribution of packets by the motes that have already forwarded these packets. Time to live counter is also implemented. It allows for forwarding the packet not much further than the closest gateway. Attacks based on feeding large numbers of packets into the network should not cause broadcast storms. It would only generate more traffic and might cause more collisions to occur.

This might be prevented by using some sort of authentication between the motes. This however is far beyond the scope of this dissertation project.

Chapter 4: Implementation

4.1 Data communication time constraints

It is very difficult to meet the time requirements which involve the drivers' reaction time as well as the very short periods of time during which vehicles moving at high speeds are able to establish a wireless link, and transfer the data using short range wireless standards.

The most important constraint is set by the drivers' reaction time. We need to inform the driver about a pedestrian or a dangerous situation on the road as early as possible, so that he/she will have time to make a decision what manoeuvre to perform.

The researchers of the University of Iowa have divided the reaction of the driver into three stages:

- accelerator release
- movement time from accelerator release to initial brake press
- initial brake press to maximum deceleration.

According to research carried out in the same University [14] using either "Iowa Driving Simulator" or real road situations, it takes nearly 2 seconds for the driver just to release the acceleration pedal and over 2 seconds to perform initial steering reaction. We need to assume that the driver needs the information to "sink in" after the warning is presented to him/her. This means that another 1 or 2 seconds are needed after the information is presented.

Concerning the way the information will be presented to the driver, before any testing is performed we need to assume that the shortest time needed to effectively communicate the information to the driver by the cars visual and/or voice interface is going to take approximately 6-7 seconds.

This means that the information about a pedestrian or a dangerous incident on the road needs to be propagated to the vehicle at least 9 seconds before the vehicle reaches the point of predicted collision. A vehicle may move up to 100km/h on a rural road, which means that every second it travels approximately 27.77 metres.

$$D_{critical} = 9 * 27.77 = 249{,}93$$

D = critical distance, which is the minimum distance that the information needs to be sent over to the car travelling at the speed of 100km/h in order to avoid a collision.

In this case, the system must be capable of sending the information to the car before it reaches the distance of approximately 250 metres before the point of predicted collision. This constraint influences only the gateway to vehicle communication. However, it is crucial to provide a system that will have a minimal detection latency and overhead of the routing protocol. As this is a real time system that is very sensitive to out of date information, we need to provide the mechanisms of delivering the data for further processing almost instantly while ensuring as long battery life as possible.

4.2 Gateway to vehicle and inter vehicle communication constraints

When designing the framework we were taking into consideration technologies available on the market, especially those used or being introduced in car industry at the moment. This includes GPS, cellular technology or WiFi.

As described above, an assumption has been made that the information will be sent to the car using a gateway and short range communication standard. We were considering the use of the following technologies- G3/UMTS, WIFI, DSRC, GPRS, WiMax as well as technologies used in FleetNet project. In the scenario where a gateway is communicating with a vehicle moving at high velocity, the two most important issues are, the range of the wireless standard and the latency of the technology used. The range of a chosen technology needs to allow the destination to be in range for a time long enough to send a complete message.

The main technology in the data link layer used to route the information between the vehicles that was taken into consideration and eventually was chosen for the present implementation is WiFi. This is motivated by the fact that more and more drivers are

using GPS and WiFi enabled handhelds in combination with road maps to successfully find the destination. Another argument that supported the choice of WiFi as a default short range standard is that it may be used for inter vehicle communication as well. There are many third OSI Layer routing experimental standards that work on top of the WiFi, like for example the “parasitic routing”, and many ad-hoc routing protocols like DSR, AODV etc.

Inter vehicle communication seems to be more problematic as the cars move at high speeds and it might not be feasible to establish the communication between the vehicles for the time sufficient to send all required data packets. This affects especially the vehicles moving in opposite directions. “Different flavours of 802.11 have a typical range of 100 metres (outdoors) (D-Link DWL-500 has a communication range between 100 to 300 metres). With simple external antenna, the range can be increased to up to 1Km. In DSRC standard, a wireless link is expected to have a maximum line-of-sight" range of 1Km.”[15] Considering the maximum range of 100 metres for some WiFi interfaces, the time that would be left for establishing the connection and sending all the data packets between two cars moving in opposite directions at the speed of 100 km/h oscillates around 2 seconds. In this case, the use of additional antenna is necessary.

4.3 The hardware

As we have decided to concentrate on the core of the system, the hardware selected for implementation covers the area of ad-hoc networks as well as wireless networking. We had to choose solutions that would allow us to scale the system up and allow for further work and adding additional functionality in the future. The hardware had also to fulfil the constraints and requirements described in previous sections.

4.3.1 Mica2 motes

In the communication framework we used sensor modules operating at 900MHz. Preliminary experiments and development of the detection algorithm considers Xbow motes as the basic element of the system.

Fig 14 Mica2 mote

Wireless Measurement System, MICA2

- 3rd Generation, Tiny, Wireless Smart Sensors
- TinyOS - Unprecedented Communications and Processing
- >1yr Battery Life on AA Batteries (Using Sleep Modes)
- Wireless Communications with Every Node as Router Capability
- 433, 868/916, or 310 MHz Multi-Channel Radio Transceiver
- Light, Temperature, RH, Barometric Pressure, Acceleration/Seismic, Acoustic, Magnetic, GPS, and other Sensors available [16]

4.3.2 Serial mote gateway

The use of MIB 510 Serial Gateway seems to be the best choice for our project. For convenience and ease of programming, a serial connection is desired. Xbow gateway is supplied together with software interfacing the RS232 port. This makes the development much easier as we don’t have to worry about handling the basic communication signalling (for example serial port driver).

Fig 15 MIB510 serial gateway

MIB510 serial gateway

- MICA2, MICAz and MICA2DOT Connector for Programming
- RS-232 Serial Interface up to 57.6Kbaud with MICA2 plugged into MIB510
- Shipped with Wall Power Supply
- Low-Voltage Detection Circuit
- MICA2, MICA2DOT LEDS mirrored on board for easy debug
- No computer parallel port is needed for programming
- Addresses and fixes the issues related to UISP programming problems, flash errors.
- Faster program downloading into the motes over serial port at 115K baud. [17]

4.3.3 Smart Roads Mote to Car gateway

Present implementation uses a regular laptop computer equipped with a wireless 802.11b interface as a gateway. It is directly connected to the serial gateway and allows for convenient execution of the gateway software, monitoring of the results as well as for debugging the system. The USB to RS232 adapter had to be used in order to connect the serial gateway to the computer. Setting up the TinyOS environment required either the use of a Linux system or a Windows platform in combination with Cygwin Linux emulator. We have chosen the second solution. The installation went smoothly and there were no problems running the system. The only problems we had were related to USB to RS232 adapter. TinyOS had problems recognizing the adapter as a serial port. However, we managed to set it up and control the MIB510 gateway using this emulated connection.

4.3.4 Car on board system

We have not managed to design and develop a complete system for the car that would allow to receive the information over a wireless link and communicate them to the driver. In the future the use of a PDA with a wireless adapter and a directional antenna should be taken into consideration.

At present we are using a wireless enabled laptop computer to receive the information from a server multicasting the location data from the gateway. The client software running on the computer displays the changes in the location of the beacon on the screen.

4.4 The software

The software part of the implementation is not only concentrated with the development of new software but also aimed at using the software already deployed. We managed to use the applications supplied by Xbow as well as expand and modify the existing code. We designed and deployed our own applications that would fit within the existing software structure and carry out it’s roles as a part of the system.

4.4.1 Motes’ software

Xbow motes use TinyOS as an operating system. TinyOS is an open-source operating system. It is designed for wireless networks and embedded systems. It has a component-based architecture which makes the process of implementation much easier and also minimizes code size, which is strongly required by severe memory constraints typical for embedded systems and especially wireless sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools. All of this components are allowed to be used changed or unchanged. They can be refined for a custom application if there is a need for changes due to required functionality of the components. TinyOS's has an event driven execution model that enables accurate power management allowing for the implementation of complicated scheduling. All this features and strong flexibility of the system fulfil most of the strict requirements set by wireless sensor networks and thus make it a very good platform for deploying distributed networks of embedded devices that have severe power, processing and communication limitations. [18]

4.4.2 Gateway's software and car's on board system

The Gateway is running a Windows system which uses Cygwin emulator that is required by TinyOS environment. The information between NesC software used to collect the data and the Java software is exchanged with the help of applications supplied by Xbow. In this project we used TOS_Base, a NesC application which is accessed by Serial Forwarder Java application. Both of the programs were supplied by Xbow. The Java application for the base station is built on top of the Serial Forwarder.

The car's on board system consists of the laptop computer with a wireless interface. The computer runs a client application in Linux environment. The application listens for multicasts sent by the base station. All updates concerning a present location of the beacon in relation to motes on the road are shown on the screen. It is important to say that not all packets received from the base station influence the changes. If the location didn't change, that is if the calculated location points to the same node in an ad-hoc network, the location information is would not be updated on the car's on board system.

4.5 The structure of an ad-hoc network – communication limitations

As we have chosen the completion of the communication framework as a primary goal. We focused most of our attention on establishing the communication between the motes. We predict that some problems may arise while testing the connectivity between the forwarding motes. This is partly due to the Fresnel zone and the position of the motes. The range of radio modules decreases due to bad weather conditions as well. This reasons imply relatively small distances between adjacent nodes on the road.

4.5.1 The Fresnel zone

The Fresnel zone is a electromagnetical phenomenon. It affects either radio or light signals. The signals get bent or diffracted from solid objects near their path. It is one

of the centric ellipsoids of revolution in wireless telecommunication. It defines volumes in the radiation pattern of a circular aperture.

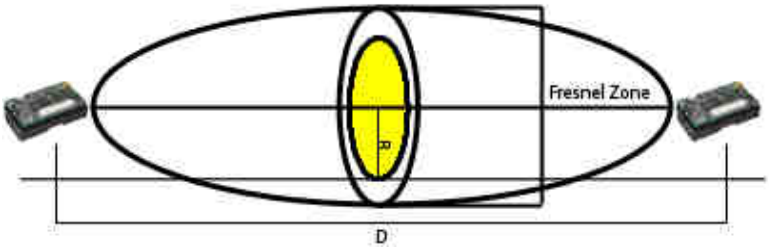

Fig 16 The Fresnel zone, [19]

As outlined in Fig. 10 our system may suffer from insufficient wireless connectivity which is caused be the phenomena referred to as the Fresnel zone. This states that the closer the antenna is placed to the ground, the more it's range decreases.

According to the performance evaluation carried out in [19], it may turn out that we will have to place the motes relatively close to each other. This may require the distance as short as 5 metres. We however aim to carry out a few different tests with different distances between adjacent motes to measure the ratio between the resolution of the system, influence of the Fresnel zone in the communication model as well as the amount of interference measured.

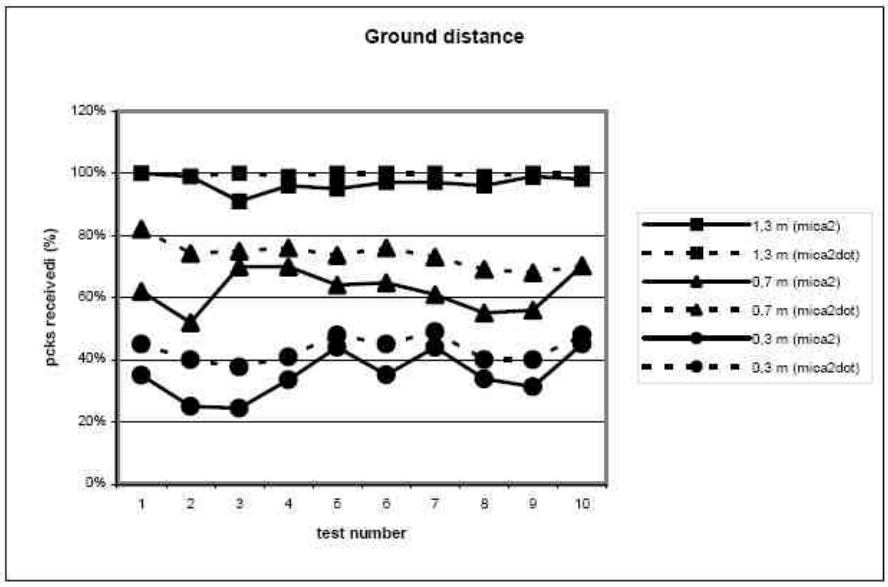

Fig 17 Influence of the sensor node's height from the ground, [19]

4.5.2 The influence of environmental conditions on the performance of radio modules

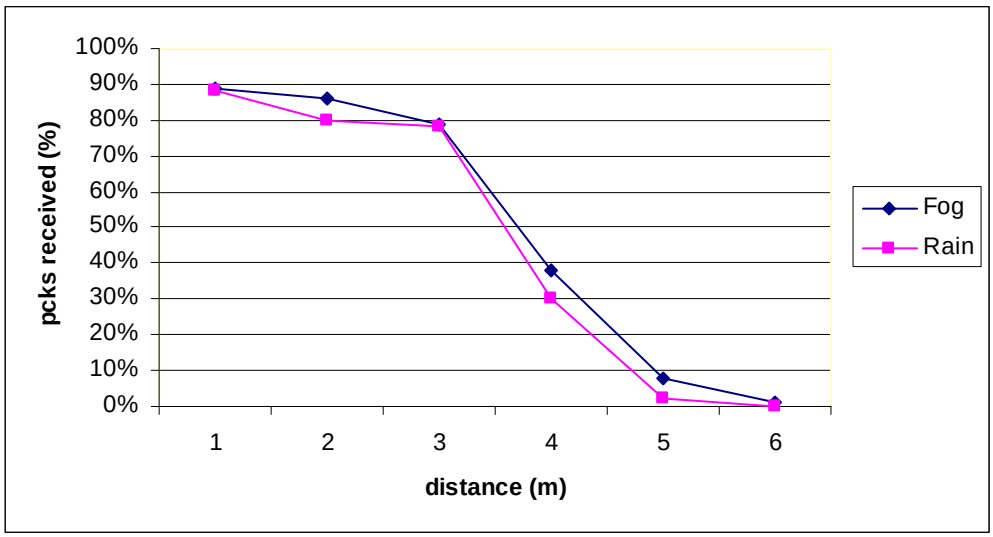

Fig 18 Mica2 motes range in the fog/rain, [19]

It is easily visible from the graph above that the range of the motes greatly decreases in the rain and fog.

These experiments justify placing the motes relatively close to each other. This however might change during the experiments. We want to measure what will be the

optimal distance between the motes. Another issue that will require some experiments is the placement of the beacon in the vest carried by the pedestrian.

4.6 RSSI measurements

The process of inferring a presence of a pedestrian is based on the strength of a signal send by the beacon mote and received by motes on the road. The strength of the received signal is easily accessible in a message structure in TinyOS.

The strength field (TOS_Msg->strength) of the TOS_Msg structure is filled with data from the radio's Received Signal Strength Indicator (RSSI) pin. The RSSI data is inversely proportional to the signal's strength.

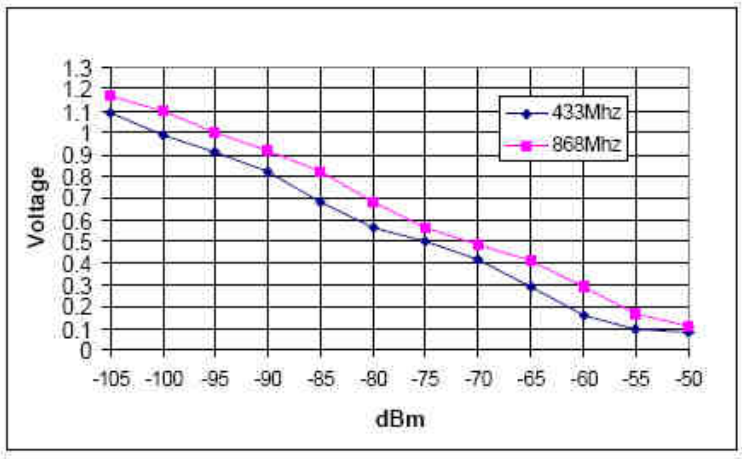

Fig 19 RSSI voltage vs. input power for CC1000 radio, [20]

CC1000 has a built-in RSSI (Received Signal Strength Indicator) giving an analogue output signal at the RSSI/IF pin. The IF_RSSI bits in the FRONT_END register enable the RSSI. When the RSSI function is enabled, the output current of this pin is inversely proportional to the input signal level. The output should be terminated in a resistor to convert the current output into a voltage. A capacitor is used in order to low-pass filter the signal. The RSSI voltage range from 0 – 1.2 V when using a 27 kΩ terminating resistor, giving approximately 50 dB/V. This RSSI voltage is measured by an A/D converter. The higher the voltage is, the lower the input signal. RSSI measures the power referred to the RF_IN pin. The input power can be calculated using the following equations:, [20]

P = -51.3 VRSSI– 49.2 [dBm] at 433 MHz

P = -50.0 VRSSI– 45.5 [dBm] at 868 MHz

The A/D converter measurements need to be referenced to the voltage of a battery in order to give trustworthy readings.

The measurements are then sent to the base station for further processing.

4.6.1 MBeacon application

The beacons' software is designed to send periodic message packets to a TinyOS broadcast address (TOS_BCAST_ADDR). Apart from the standard fields, the message contains additional information necessary to precisely compute the location of the beacon in relation to other motes. Figure 20 depicts the exact structure of a packet that is forwarded within the ad-hoc network, after the fields of the packet are filled out with data by receiving forwarding node. Please note that the following example concerns the packet that is carried by forwarding nodes to the base station. The frame structure is the same as the frame of the packet created by MBeacon application. The only difference is that MBeacon program fills out only sourceMoteID, sendingTime and the default fields of a TOS_msg data structure.

dest	handlerID	groupID	msg len	sourceMoteID	sendingTime	receiveNodeID	msg_strength	dBm	packetID	TTL
2byte	1byte	1byte	1byte	2byte	4byte	1byte	2 byte	2 byte	2byte	1byte
FF FF	01	81	0E	8E 01	64 16 98 00	01	1D 00	84 13	A5 5B	3C

Fig 20 MBeacon message structure

The following fields are contained within the frame:

- dest - the destination of a packet (default)
- handlerID – the message handler ID (default)
- groupID – the ID of the group within which a packet is sent (default)
- msg len - the length of the message (default)
- sourceMoteID – the ID of the beacon
- sendingTime – the timestamp of the packet. It is a timestamp of a particular beacon at a particular moment in time and used to distinguish between the packets received from the same beacon.

- receiveNodeID – the ID of the node that has received the signal from the beacon.
- msg_strength - the strength of the signal. This variable is calculated by one or more motes placed on the road and represents the strength of the signal of the message sent by the beacon and received by the mote. Please note that this value does not represent the actual RSSI value, but raw ADC readings and is used for debugging.
- dBm - is actual RSSI value referenced to the voltage of each of the receiving nodes separately
- packetId – is a unique packet identifier that is used to distinguish between different packets. It is assigned at a mote that receives the signal from the beacon and forwards the information. It is then used in routing to avoid forwarding duplicate packets and broadcast storms caused by overhearing and forwarding the same packets multiple times.
- TTL – a time to live field determines the range within which the packet can be forwarded and is limited to 60 hops only.

MBeacon application consists of two files. One of the files is a configuration file that wires all the necessary interfaces to the implementation. The MBeaconM.nc file implements the software.

The messages are sent every 500ms. TimerC is the component that provides necessary interface.

4.6.2 MyRoute application

The mote network is not a completely ad-hoc. The motes that are supposed to be placed along the road seem to have more features of a fixed network rather than a complete ad-hoc and chaotic group of communicating nodes.

This allows to ignore most of ad-hoc routing protocols and turn toward a simple routing solution that will reliably forward packets toward the base station with as little latency as possible.

The functionality of MyRoute application is very simple. It is installed on every mote, apart from beacon nodes and the mote receiving the signals on the base station. It waits for broadcasted messages from beacons and other motes.

As soon as a message from beacon arrives at the mote on the road, MyRoute application calculates the strength of the signal and enters a value in the msg_strength field of the packet. The strength of the signal is a part of a default TOS_msg structure.

```
typedef struct TOS_Msg
{
        /* The following fields are transmitted/received on the radio. */
 uint16_t addr;
 uint8_t type;
 uint8_t group;
 uint8_t length;
        int8_t data[TOSH_DATA_LENGTH];
        uint16_t crc;

 /* The following fields are not actually transmitted or received
  * on the radio! They are used for internal accounting only.
  * The reason they are in this structure is that the AM interface
  * requires them to be part of the TOS_Msg that is passed to
  * send/receive operations.
  */
 uint16_t strength;
 uint8_t ack;
 uint16_t time;
 uint8_t sendSecurityMode;
```

```
  uint8_t receiveSecurityMode;
} TOS_Msg;
```

[21]

Even though the field containing the data with the strength of the signal is easily accessible the readings are not real dBm values. These are raw ADC readings and need to be referenced to the battery voltage.

```
async event result_t ADC.dataReady(uint16_t batterydata){
    float x;
    CLEAR_BAT_MONITOR();
    x = (float)batterydata;
    x =  125235 / x ;
    voltage = (uint16_t ) x ;
    return signal Battery.dataReady(voltage);
 }
```

The acquired voltage can be then applied to the following equation:

MyVrssi = ((volt/100) * Msg->strength)/1024;

Having the real RSSI values, we can calculate the actual strength of the received signal in dBm:

Dbm = (51.3 * MyVrssi) + 45.5;

dest	2byte	handlerID	1byte	groupID	1byte	msg len	1byte	sourceMoteID	2byte	sendingTime	4byte	receiveNodeID	1byte	msg_strength	2 byte	dBm	2 byte	packetID	2byte	TTL	1byte
FF FF		01		81		0E		8E 01		64 16 98 00		01		1D 00		84 13		A5 5B		3C	

Fig 21 Data packet after adding in the signal strength value

The IDs assigned to beacons are greater than 255 (sourceMoteID field - byte). Thanks to assigning a different addressing space to motes used for forwarding the messages and beacons, receiving nodes can easily distinguish between the signals arriving from beacons and packets broadcasted by other motes used for forwarding the data.

The addressing is not structured and it has no influence on the routing protocol. Using a simple routing protocol that is based on broadcasts has many advantages. The protocol has nearly no overhead. The messages that would have to be generated and flooded to the network in order to find the route to the destination would generate additional traffic and use more bandwidth than the present implementation.

Broadcasting may however be dangerous as it creates more traffic than unicast, because messages are addressed to all nodes. In order to prevent broadcast storms, two countermeasures have been implemented. Forwarding nodes buffer the last 30 IDs of received packets and every time they receive a new packet, they look for recently received IDs in the ID table of packets, that have already been received, in order to check whether the message has already been forwarded. If the packet is already in the ID table, the packet is dropped and is not forwarded again. This mechanism prevents the nodes from sending the same messages infinitely, generating more and more traffic, using more energy and slowly bringing the whole system down. Of course some adjustments may have to be applied. It may turn out that the number of buffered packet IDs may have to be increased. This may be caused by many beacons moving in the same direction, in close proximity (a group of people).

Another countermeasure implemented in the routing protocol is a TTL field. It limits the life of every packet to 60 hops. This mechanism prevents the packets to be forwarded too far. As we assume that the base stations will be placed on the road every 200 – 300 metres, the packets shouldn’t be forwarded too far.

4.6.3 Base station mote NesC software

For testing purposes the TOSBase application has been used. It captures all the packets that it can hear on the radio interface and reports them back to the serial port. It also forwards all incoming UART messages out to the radio interface. The application has been used in combination with net.tinyos.tools.Listen application. It displays all messages received on the radio interface and forwarded to UART. [22] Before running the Listen software, the environment variable MOTECOM needs to be set up. It tells the java Listen tool (and most other tools as well) which packets it should listened to. The baud rate says to listen to a mote connected to a specific serial

port at a specific interval. The baud rate is the specific type of mote. For the mica and mica2dot motes, the baud rate is 19200, for the mica2 it is 57600 baud.

4.6.4 Serial Forwarder

Serial Forwarder is an application that is used to read packet data from a serial port and forward it over an Internet connection. It allows other programs to be written to communicate with the sensor network over the network. If Serial Forwarder is running on the same machine as the client application, the “localhost” port is used. The use of Serial Forwarder is simple. It is accessed by programs by creating a socket connection on a designated port. This allows for very easy creation of a wide range of different applications that may easily access the data gathered by ad-hoc network of interconnected motes.

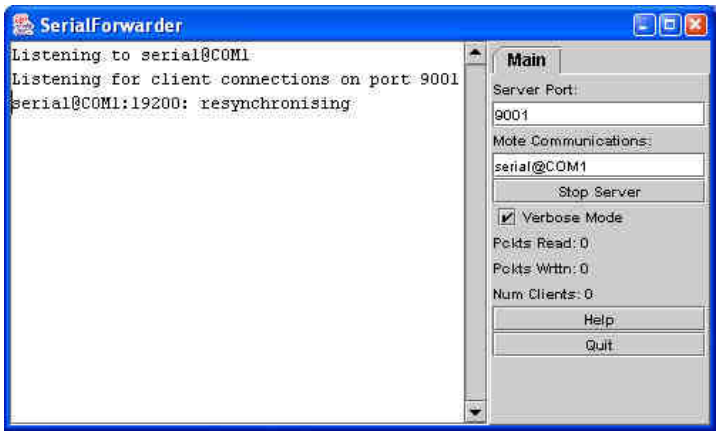

Fig 22 Serial Forwarder program

Serial Forwarder does not display the packet data itself. Instead, it listens for network client connections on a given TCP port (9001 is the default), and simply forwards TinyOS messages from the serial port to the network client connection, and vice versa.[22]

4.6.5 Base station’s Java software

Base station Java software communicates with the motes programmed in NesC, using Serial Forwarder application described above. As soon as Serial Forwarder creates a socket, the application simply makes a TCP connection on the port 9001 using “localhost” as a destination address. Serial Forwarder allows for establishing

connections over network, however for our solution remote access to data gathered by the motes is not necessary.

To easily read the messages, which are forwarded by Serial Forwarder, a Mig tool had to be used. The messages are forwarded to the created socket in raw byte form and need to be parsed before accessing. Mig is a Message Interface Generator for nesC which allows for automatic creation of Java classes that are used to access the data encoded in each packet. To generate a java class used for accessing the data a structure describing information contained within the packet and stored in a header *.h file must be used. The following structure was used to generate the java class:

```
typedef struct MyData{
        uint16_t sourceMoteID;   //beacon ID
         uint32_t sendingTime;    //beacon timestamp
        uint8_t receiveNodeID;   //id of mote that received signal from beacon
        uint16_t msg_strength;
        uint16_t Dbm;
        uint16_t packetID;
        uint8_t TTL;

} MyData;

enum{
    AM_MYDATA=2 // represents the type of a message
};
```

The command:

```
mig -target=mica2 -o MyData.java -java-classname=net.PDetection.MyData java MyRouting.h MyData
```

generated necessary Java class file and enabled access to forwarded data without the necessity of parsing byte messages.

The Base station application uses the same mechanism preventing it from storing duplicate packets as the routing protocol. It checks the ID of each packet that is received from the Serial Forwarder. If the packet has not been received previously, it is stored together with the ID of the beacon that the message originated from, the ID of the mote that received the packet from the beacon, the strength of the signal (RSSI value), and the timestamp of the beacon:

```
public int[] packetIDTable = new int[30];

public short[][] receiveNodeIDtab = new short[10][20];

public long[][] timestamptab = new long[10][20];

public int[][] dBmtab = new int[10][20];
```

If timestamp contained within received packets increases twice, the beacon location calculation is triggered. The base station calculates the position of the beacon using different number of received packets, depending on how many nodes on the road received the signal from the beacon.

As the beacon sends the information every 500ms, the initial time from which the beacon is detected for the first time, till the moment when the position is calculated equals 1500ms + routing latency. Routing latency is a variable in this equation, but it should not influence the overall performance of the system. We will measure the initial and overall latency by carrying out tests.

The base station software does not only calculate the location of the beacon, it also sets up a UDP server and multicasts the information.

```
InetAddress group = InetAddress.getByName("230.0.0.1");

DatagramPacket packet = new DatagramPacket(buf, buf.length, group, 4446);
```

It sends the information to the multicast group 230.0.0.1 on the port 4446, so any client within this group listening on a designated port can receive the data. We have used UDP for wireless link, because due to harsh time constraints there would be insufficient time to establish a TCP connection and send the data. TCP is a connection oriented protocol that provides mechanisms ensuring data delivery, but it has relatively high overhead. It uses a mechanism called three-way-handshake to

establish a connection and is not suitable for the environment where instant connections are required. UDP on the other hand is connectionless and does not need to establish the connection. The client needs to listen on the port and be in a defined multicast group to receive the message.

4.6.6 Client's Java software

Current implementation includes client's software for receiving location updates on the car's on board system. Java client application listens on port 4446 for incoming packets. If a packet is received, the client checks if a particular beacon changed it's location since last update. If the location is still the same, the data displayed does not change. If however, the beacon is reported to be at a different location, the client software displays updated information.

Chapter 5: Evaluation

5.1 Evaluation goals

We carefully chose different types of tests to measure the overall performance of the communication framework. We aimed at identifying the parts of the system that perform well enough to be a starting point for further development and could be used in the future implementation after slight modifications. We also kept in mind that the first tests should evaluate the suitability of either hardware or software and measure the optimal configuration of the system.

There are technical papers evaluating the performance of Mica2 motes' radio modules in various environmental conditions, either indoors or outdoors, so we were able to specify the boundaries within which the tests should be carried out. That included the physical layout of the motes and the configuration of the software.

Apart from testing the suitability of the hardware used, the tests focused on the developed routing protocol and it's performance in relatively large ad-hoc network consisting of up to 13 motes and several beacons. We also wanted to test the Java software developed for the base station as well as the client's software created for car's on board system.

5.2 The Tests

We carried out several different tests in varying conditions and communication framework's configurations. The variety of different tests helped us to understand whether the technology used is mature enough to be used in a time constrained real time system like Pedestrian Detection System.

The tests can be divided into two separate groups:

- Extensive indoor testing
- Outdoor tests and evaluation

Even though outdoor testing is better justified than the tests that we carried out inside buildings, we think that the indoor testing does give a many valuable information about the behaviour of the framework in the presence of obstacles and distorted signal. We think that to some extent the indoor tests display the performance better than outdoors tests as the indoor environment makes the radio signal distort and reflect from walls and other obstacles and thus measure the resilience of the hardware to distorted messages. It also shows how well the routing protocol manages to forward the information in highly congested environment where many nodes overhear the same signal.

5.2.1 Indoor testing

A large variety of indoor tests has been carried out. The first tests aimed at measuring the resilience of the routing protocol in highly congested but small network. The tests that we managed to carry out later aimed at measuring estimated accuracy and performance of the communication framework.

5.2.1.1 First tests used for debugging the routing protocol

The first test involved the use of two beacons sending signals every 500ms. The test was carried out in a single room with four motes acting as routers and one mote acting as a base station. The test lasted for two minutes and gave us valuable information about the resilience of the routing protocol to interference caused by close proximity of the motes as well the accuracy of measured RSSI signals.

During the test, the motes acting as routers were placed in different places in the room. The distance between adjacent routing motes was approximately 2 metres. The beacons were placed close to two different motes at a distance of approximately 20cm.

It is worth to mention that this test was carried out during the development phase and did not involve fully completed software. It proved that our approach to routing was successful as we haven’t noticed packet loss, and the delay in displaying up to date

information. Only when the routing nodes were placed very closed to each other (less than 50cm) we have noticed the delay in the delivery of the packets and some of them being lost.

5.2.1.2 Indoor testing environment and assumptions

To make it possible to compare different indoor tests, we assumed the same layout of the network (a straight line). We also decided that the test should be somewhat static. This means, that the beacon carried by a “pedestrian” would be sending signal from the places set up in advance – the person carrying a beacon would stand beside each node and between each two adjacent routing motes for the time of approximately 10 seconds. This would give an acceptable set of data, useful in further analysis and comparison. The time of 10 seconds allows the beacon to send 20 messages (a message is sent every 500ms) and gave us approximately 6 location readings for each position. I am using the word “approximation” because we did not measure the exact time that the person stood in one place. The amount of time however, does not influence the results of the tests. The important fact for us, was to get at least 5 location values calculated at the base station, that would allow us to compare the results.

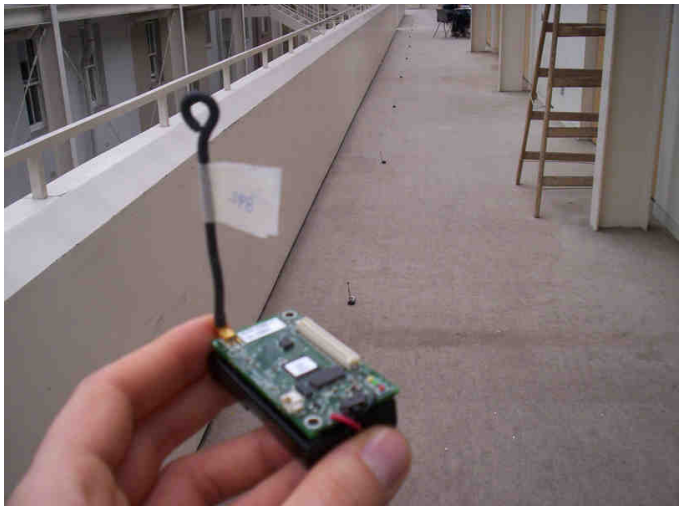

Fig 23 The mote used as a beacon and routing motes in the background

Fig 24 A general layout of the forwarding nodes

A laptop computer was used as a base station. It was running Cygwin emulator. The mote with TOS_Base software installed was connected to the MIB510 gateway. The gateway was connected to the computer via RS232 port, through a USB to RS232 adapter. The data was received by a Serial Forwarder application that forwarded the messages to the dedicated base station software.

Fig 25 The base station with the MIB510 gateway connected to it

The base station software was processing received and buffered messages and displaying the inferred location of a beacon at a given time. The data was logged to a

text file at the same time to make it easy to compare different sets of data from different experiments.

Fig 26 The base station to the left, the ad-hoc network and the person acting as a pedestrian carrying a beacon

It is easily visible from the pictures, that the corridor used for testing was very narrow. The motes were placed close to the wall to the left, which reflected and distorted the signal. The environment greatly influenced the results.

It is important to say that the addressing of the motes does not influence the routing protocol. The differences in the location of particular forwarding motes are relevant only to the base station which needs to know the exact location of the motes for the location information to be of any value. However, for the routing protocol, addressing is not important.

The IDs of the forwarding nodes varied between 10 and 21. The beacon used for testing had the ID = 398. For the convenience of the analysis of the acquired data, the motes' IDs were increasing as the distance from the base station increased. The layout of the motes with particular IDs was as follows:

- Base station – 10 – 11 – 12 – 13 – 14 – 15 - 16 – 17 – 18 – 19 – 20 - 21 (indoors – 3 metres distance between the motes)
- Base station – 10 – 11 – 12 – 13 – 14 – 15 (indoors – 5 metres distance between the motes)

- Base station – 10 – 11 – 12 – 13 – 14 – 15 (indoors – 7 metres distance between the motes)
- Base station – 10 – 11 – 12 – 13 – 14 – 15 (outdoors – 5 metres distance between the motes)
- Base station – 10 – 11 – 12 – 13 – 14 – 15 (outdoors – 7 metres distance between the motes)

5.2.2 Initial test involving 12 motes (3 metres between adjacent nodes)

The first major test involved the use of relatively large network of 12 motes acting as routers, each of them placed in a straight line in a narrow corridor. There was only one mote acting as beacon, sending a signal every 500ms. All other motes were acting as routers, and were forwarding the information with a measured strength of the signal.

The first test with 12 mote network layout was a dynamic test, ensuring that the whole system is gathering and processing the information. During this test, a person walked along the line of motes carrying a beacon. The speed of a person may be described as a medium speed walk. We don't regard this test as a valuable source of information as it was hard to know the real location of a person in relation to the location results displayed by the base station. The person was walking starting from a close proximity of the forwarding node with the ID = 21, which was the furthest node from the base station.

The results are as follows:

beacon: 398 close to node: 21 - 53.91

beacon: 398 close to node: 20 - 60.42

beacon: 398 close to node: 19 - 71.62

beacon: 398 close to node: 19 - 71.3

beacon: 398 close to node: 19 - 60.31

beacon: 398 close to node: 21 - 66.46

beacon: 398 close to node: 19 - 60.31

beacon: 398 close to node: 18 - 53.46

beacon: 398 close to node: 17 - 62.54

beacon: 398 close to node: 19 - 63.5

beacon: 398 close to node: 18 - 62.22

beacon: 398 close to node: 14 - 72.26

beacon: 398 close to node: 14 - 70.98

beacon: 398 close to node: 14 - 70.98

beacon: 398 close to node: 15 - 53.44

beacon: 398 close to node: 13 - 62.81

beacon: 398 close to node: 14 - 72.42

beacon: 398 close to node: 14 - 67.64

beacon: 398 close to node: 12 - 68.12

beacon: 398 close to node: 11 - 63.66

beacon: 398 close to node: 13 - 66.93

beacon: 398 close to node: 14 - 75.76

beacon: 398 close to node: 11 - 60.63

beacon: 398 close to node: 11 - 60.63

beacon: 398 close to node: 11 - 70.67

beacon: 398 close to node: 11 - 70.67

beacon: 398 close to node: 13 - 80.43

beacon: 398 close to node: 11 - 72.9

The results represent the ID of a beacon, the ID of a forwarding node that is in closest proximity to the beacon (according to RSSI) and the value of a measured strength of the signal. Even though the results don’t seem to be very accurate, the largest reported inaccuracy in forwarding motes’ IDs reported was 2. Considering the fact that the motes were 3 metres apart, we managed to achieve 6 metre accuracy. As mentioned before, it was hard to estimate the actual position of a person. However,

we estimated that the location information were 2 seconds out of date. As the new location information are calculated and displayed every 1.5 seconds, this has introduced additional approximate 2 metre error. Eventually, we have received 8 metre resolution which still is acceptable.

The error would be much greater if we considered a person who was moving much faster. To reduce the error some sort of reasoning might be introduced in the base station's software. It could predict the direction of movement of a person as well as take into consideration the fact that the data received by the base station is slightly out of date.

5.2.3 Indoor test involving 12 motes (3 metres between adjacent nodes)

The second of indoor tests was carried out in a similar conditions. It involved the same amount of forwarding motes and one beacon broadcasting the signal every 500ms. This test however was more static than the first one. This time the person acting as a pedestrian carried the beacon to the location set in advance, which was either the spot next to one of the motes or the space between the motes.

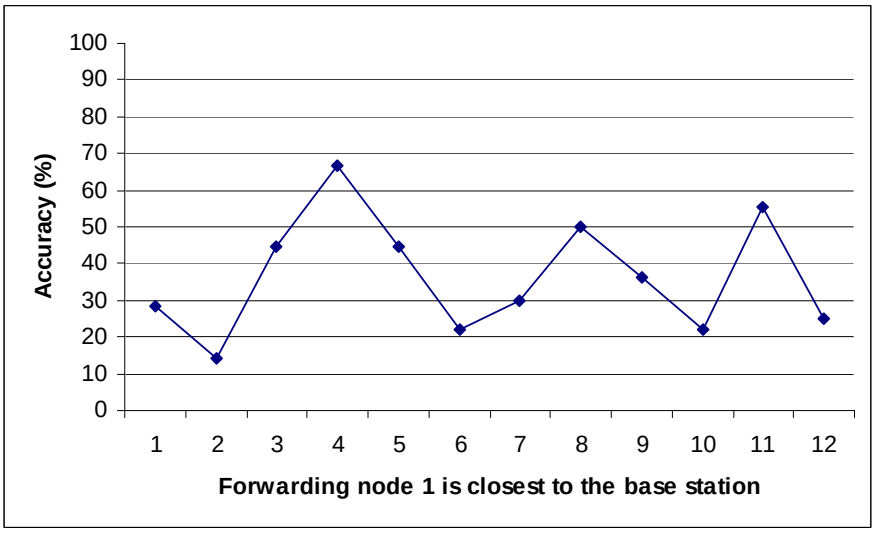

Fig 27 Measured accuracy of the estimated locations

The beacon was not turned on from the beginning of the test. The person carrying a beacon would go to the designated location and turn the beacon then for the period of approximately 10 seconds and was moving forward to the next location after turning

it off. The person was stopping next to each of the nodes as well as between adjacent nodes. The process gave us 23 different sets of measurements.

The accuracy is defined as a percentage of readings that were displayed by the base station that were correct in relation to the actual position of a beacon among all readings carried out at a particular location.

The other set of tests that was carried out with the beacons between the adjacent forwarding motes made the same assumption with only one difference. The measurement was considered accurate if the location information shown by the base station was identifying either of the adjacent nodes as a correct location of the beacon.

The measured accuracy was not perfect. According to our expectations, the accuracy of estimated location was not high. It is displayed on the graph in Fig 27.

The average accuracy of sensing the location of a beacon was an unimpressive value of 36,64%. This was calculated on the basis of 213 estimated locations during the whole experiment. This means that during the whole experiment the beacon has sent 639 packets containing it's timestamp and ID. We do not know how many packets were actually received by the base station, as only the information from the packet with the strongest strength of the signal value was displayed.

As you can see from the graph, the system did not perform very well, when it comes to estimating the exact location. This was not a surprise. Before the experiment we assumed; taking into consideration the results of many tests that were carried out by authors of different technical papers describing the communication performance of Mica2 motes; that the optimal distance between the motes should oscillate around the value of 5 metres between adjacent nodes. Forwarding nodes in our experiment were only 3 metres apart, meaning that considering the distance over which the motes can communicate as well as the test environment that "encouraged" the propagation of the radio signal, we can safely assume that all nodes in the network were receiving every signal send from a beacon every 500ms.

If we consider that all the motes were receiving the signal, it means that all 12 motes were measuring the RSSI and filling out all the fields of the routing message and sending the message at the same time. This was definitely a highly congested environment. We could expect the decrease in the routing performance. However, we

have not noticed any degradation at all. Even when the beacon was at the very far end of the network, the location information was displayed on average every 1.5 second giving the information about a beacon being somewhere at the far end of the network. The accuracy was only 36.64%, but we need to remember that we considered a measurement accurate only when the location displayed by the base station's software was actually displaying the ID of the mote that the beacon was closest to. When the measurement was taken with the beacon between two adjacent nodes, the measurement was considered accurate only when the displayed location information shown the ID of either of the closest motes. To prove it we display the measurements taken when the beacon was at the far end of the network:

beacon:	398	close	to	node:	21	-	55.41
beacon:	398	close	to	node:	18	-	74.91
beacon:	398	close	to	node:	19	-	67.75
beacon:	398	close	to	node:	17	-	74.6
beacon:	398	close	to	node:	21	-	53.05
beacon:	398	close	to	node:	21	-	52.73
beacon:	398	close	to	node:	20	-	65.47
beacon:	398	close	to	node:	20	-	66.1
beacon:	398	close	to	node:	20	-	66.26
beacon:	398	close	to	node:	20	-	66.26
beacon:	398	close	to	node:	18	-	70.03
beacon:	398	close	to	node:	20	-	65.47

The exact accuracy is very poor and has the value of 36.64%. However, as you can see the maximum error in location estimation was 4 motes (4^{th} reading), which equals 12 metres. We find this result acceptable due to a very noisy environment.

Another interesting finding is that the accuracy of the measurements of the strength of the signal degrade in the presence of nearby obstacles. As you can see in graph in Fig 27, the performance accuracy decreased strongly around the 1^{st} and 2^{nd}, 6^{th} and 7^{th}, 10^{th} mote. This is due to the fact, the there were obstacles in the closest proximity

of the motes. The obstacles can be seen on the pictures on the right hand side in Fig 23, 24 and 26.

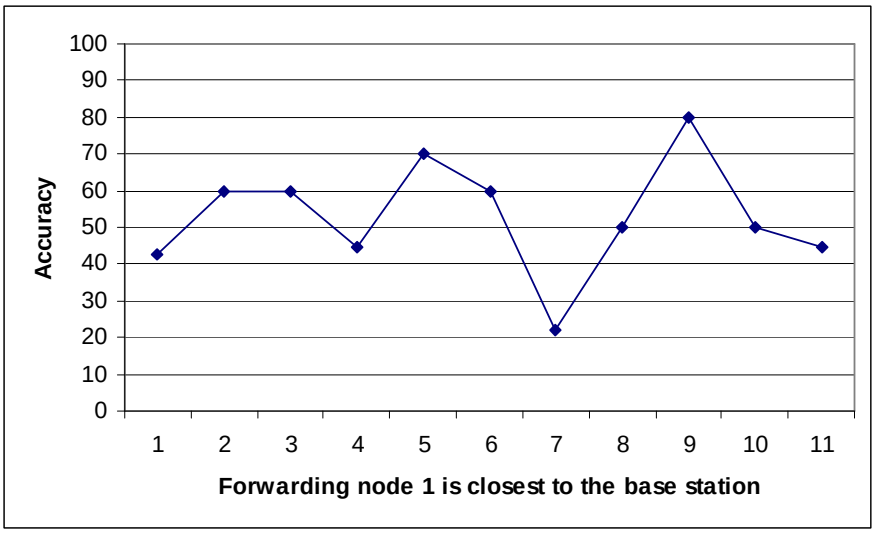

Fig 28 The accuracy with beacon placed between adjacent forwarding nodes

We considered a reading accurate when the displayed information concerned either of the IDs of the adjacent nodes.

5.2.4 Indoor test involving 6 motes (5 metres between adjacent nodes)

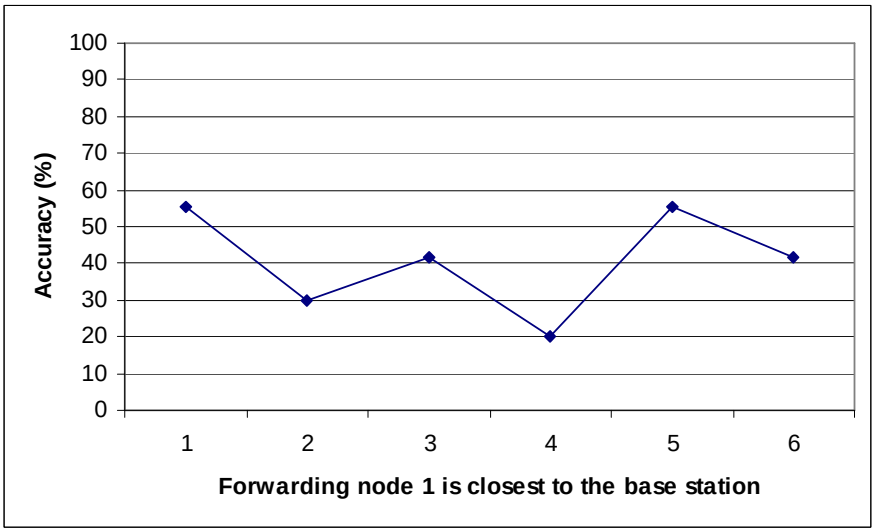

Fig 29 Measured accuracy of the estimated locations

The next test was carried out in similar conditions and in the same way as the previous one. The only change in the set up was the distance between adjacent forwarding nodes and the number of the motes involved in the test.

As we predicted, the overall performance in estimating the accurate location of the beacon increased by 4.1% and was over 40% (40.74%).

Due to a slight change in the configuration the overall performance has increased. We can still notice the degradations in the performance near the obstacles. This supports the statement that we made during the first experiment.

The measurements of the location when the beacon was in close proximity of the last node in the network, gave acceptable results again.

beacon:	398	close	to	node:	15	-	55.69
beacon:	398	close	to	node:	15	-	55.37
beacon:	398	close	to	node:	15	-	55.37
beacon:	398	close	to	node:	12	-	76.39
beacon:	398	close	to	node:	14	-	73.56
beacon:	398	close	to	node:	11	-	80.31
beacon:	398	close	to	node:	13	-	76.54
beacon:	398	close	to	node:	12	-	74.5
beacon:	398	close	to	node:	15	-	58.35
beacon:	398	close	to	node:	14	-	70.58
beacon:	398	close	to	node:	14	-	70.58
beacon:	398	close	to	node:	15	-	57.26

The maximum variation in the displayed results was 3 nodes this time, which results in 15 metre inaccuracy. The odd reading is displayed in yellow.

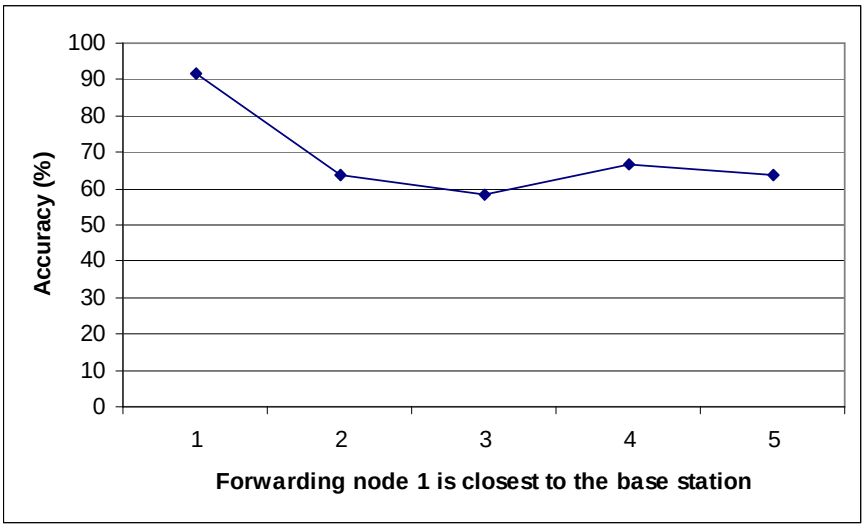

Fig 30 The accuracy with beacon placed between adjacent forwarding nodes

Again we considered a reading accurate when the displayed information concerned either of the IDs of the adjacent nodes.

5.2.5 Indoor test involving 6 motes (7 metres between adjacent nodes)

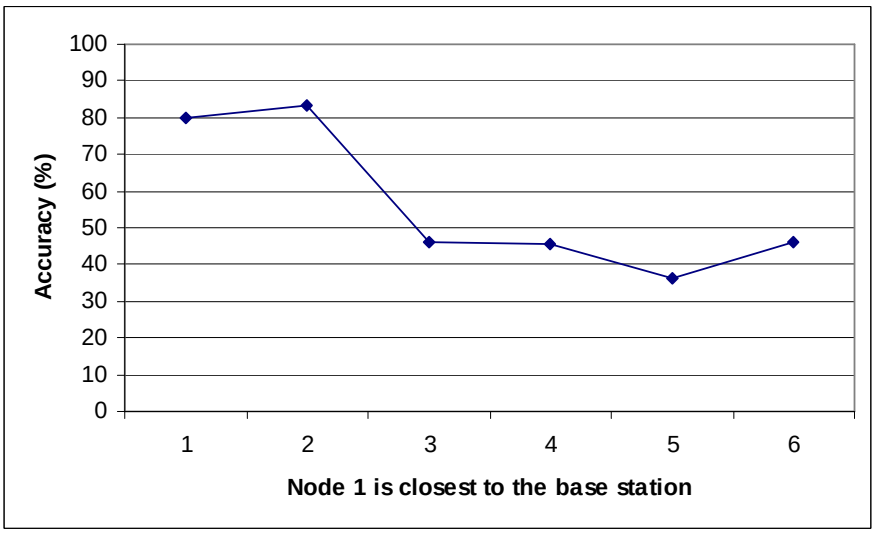

Fig 31 Measured accuracy of the estimated locations

The next test was carried out in a very similar conditions and in the same way as the previous one. The only change in the set up was the distance between adjacent forwarding nodes and the number of the motes involved in the test.

The overall performance in estimating the accurate location of the beacon increased by a 15.5% and averaged at 56.24%.

It’s easily visible that the readings are much more stable this time and oscillate around the value of 50%. We consider this an acceptable level. We need to mention that the maximum error was 3 nodes, which resulted in 21 metre inaccuracy. The odd reading is marked in yellow:

beacon:	398	close	to	node:	14	-	69.25
beacon:	398	close	to	node:	15	-	70.97
beacon:	398	close	to	node:	15	-	71.6
beacon:	398	close	to	node:	15	-	56.56
beacon:	398	close	to	node:	14	-	65.66
beacon:	398	close	to	node:	14	-	65.5
beacon:	398	close	to	node:	15	-	56.71
beacon:	398	close	to	node:	13	-	75.88
beacon:	398	close	to	node:	12	-	77.69
beacon:	398	close	to	node:	14	-	75.35
beacon:	398	close	to	node:	15	-	62.79
beacon:	398	close	to	node:	15	-	58.89
beacon:	398	close	to	node:	14	-	69.1

As in previous experiments; we considered a reading accurate when the displayed information concerned either of the IDs of the adjacent nodes.

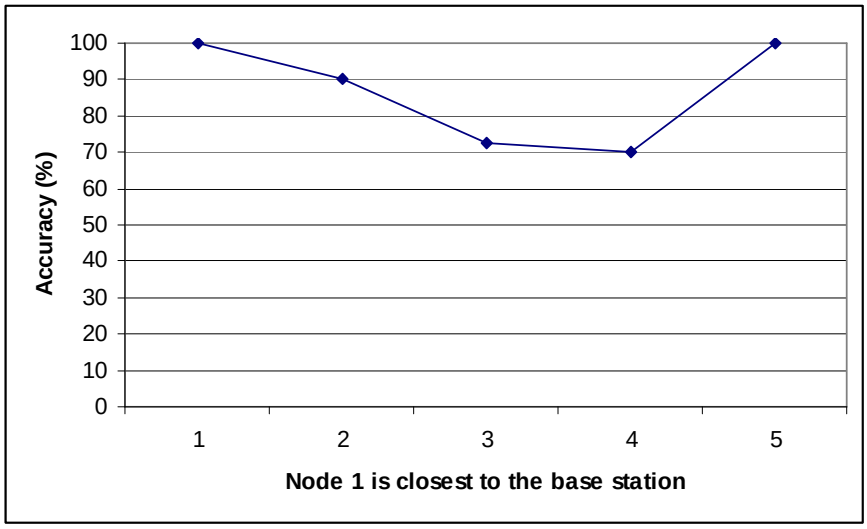

Fig 32 The accuracy with beacon placed between adjacent forwarding nodes

5.2.6 Outdoor testing

After carrying out extensive indoor testing, we decided to move the network to the environment which would be similar to real life conditions. We were hoping to get more consistent results in open air environment without obstacles near the motes that could reflect and distort the signals.

5.2.6.1 Testing environment and assumptions

Outdoor tests were carried out on the flat surface, without any obstacles in close proximity of the motes. The conditions of the test were very similar to indoor tests and involved one beacon and forwarding nodes. The layout of the forwarding nodes was a straight line.

Again the base station using exactly the same hardware and software as in the indoor tests was placed at the edge of the network of forwarding nodes.

To make it possible to compare the results of indoor and outdoor tests, we agreed to make the assumption the test would be carried out statically. Once more we assumed that the beacon would be turned on and off in fixed locations so that we could collect sufficient amount of data for comparison.

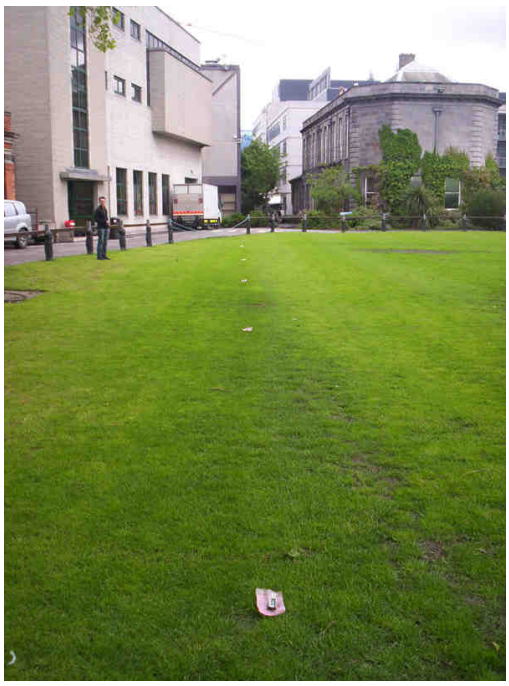

Fig 33 The layout of the forwarding motes

We decided to carry out two different series of measurements- one involving forwarding motes placed at the distance of 5 metres from each other, and the second one with adjacent nodes being 7metres apart.

5.2.6.2 Outdoor test involving 6 motes (5 metres between adjacent nodes)

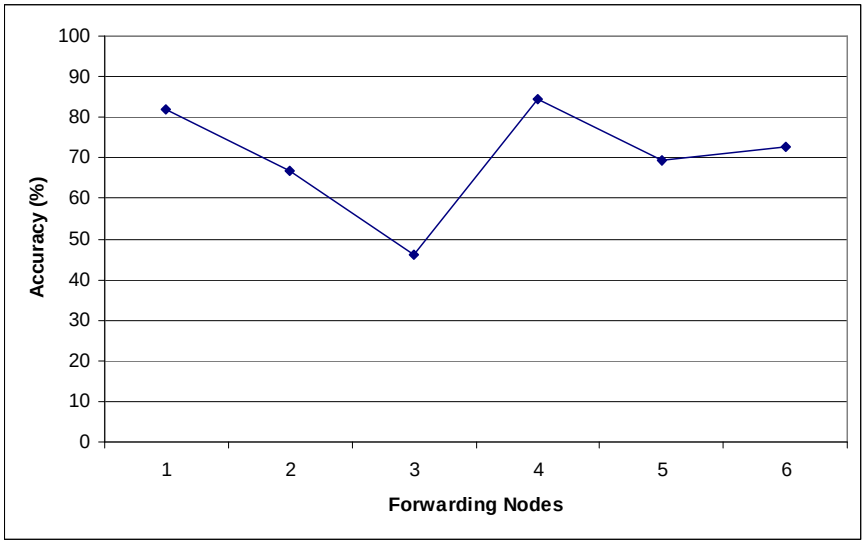

Fig 34 First set of outdoor measurements

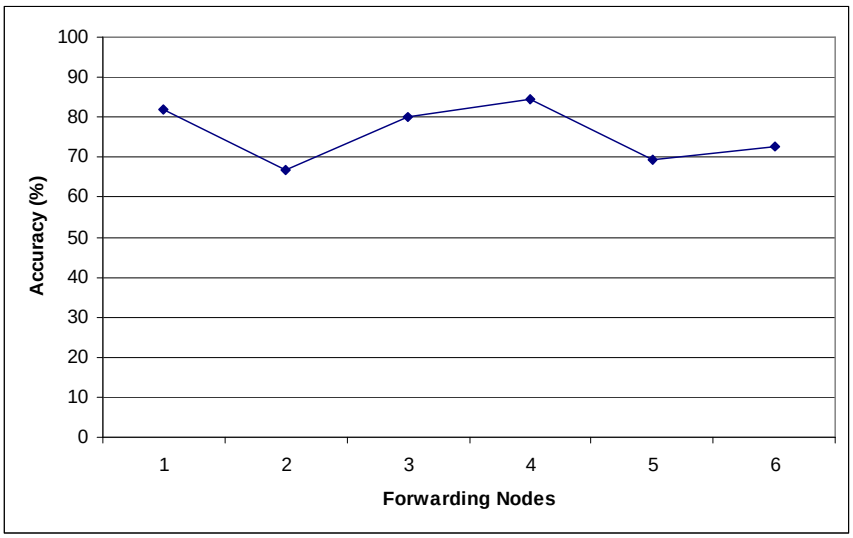

Fig 35 Second set of measurements with replaced node nr 3

The first of the test involved the distance of 5 metres between the forwarding motes.

As you can see on the graphs in Figures 32 and 33, we have noticed a greatly decreased efficiency of estimating the location around the node number 3. The Fig 33 shows the second data of measurements around node 3 when it was replaced with another unit.

The overall accuracy in estimating an exact location of a beacon was 75.84%. We consider this value a very good result. It's worth mentioning that the inaccuracy in estimating the location was a two node error, which results in 10 metre inaccuracy. Out of 138 different measurements only 1 was exceeding this boundary.

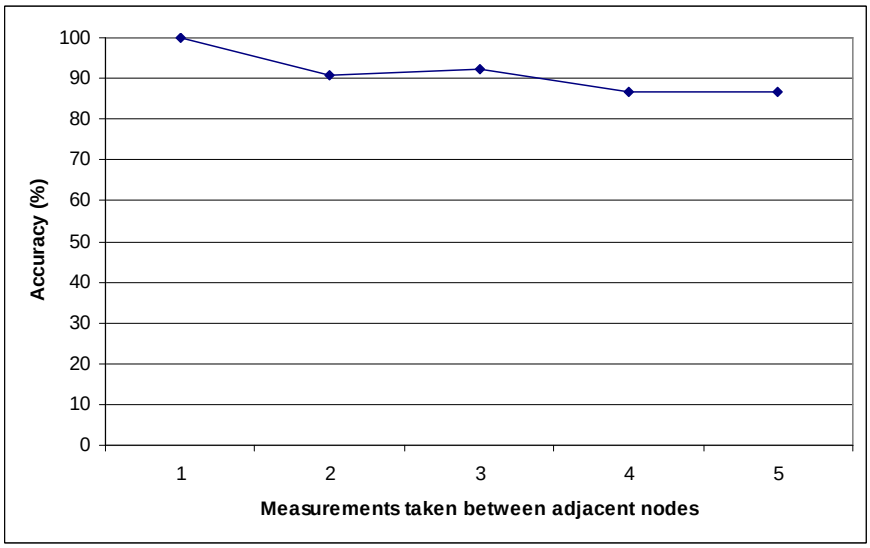

Fig 36 Measurements taken between adjacent forwarding nodes

Similarly to indoor experiments we decided to take the estimated location measurements between the motes. The measurement was considered accurate when the displayed information concerned either of the IDs of the adjacent nodes. The average accuracy was very high 91.31%.

5.2.6.3 Outdoor test involving 6 motes (7 metres between adjacent nodes)

The final test tested the accuracy of estimating the location when the adjacent forwarding nodes were 7 metres from each other.

The overall performance of the system was 72.63%, while the maximum error in the estimation was two nodes, which equals 14 metres.

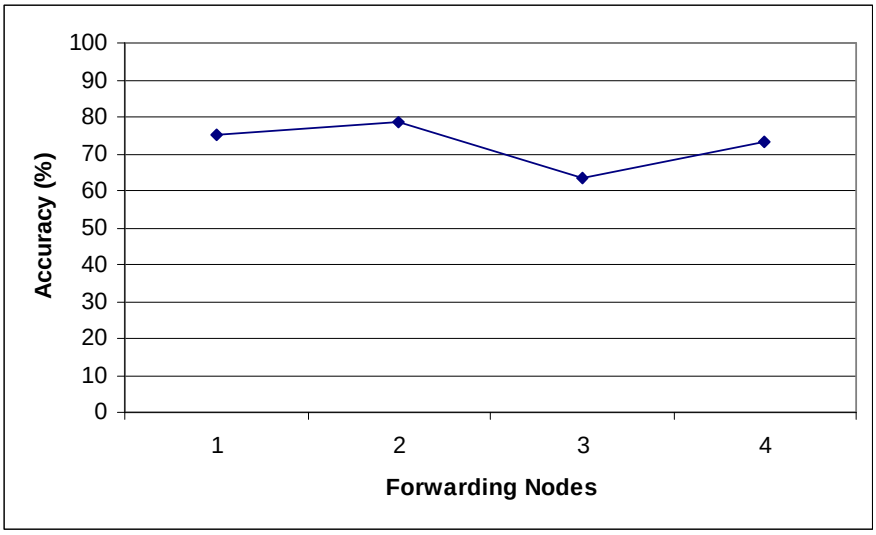

Fig 37 The measurements of the accuracy of estimation - 7 metre distance between adjacent nodes

The measurement taken when the beacon was placed between two adjacent nodes was considered accurate when the displayed information concerned either of the IDs of the adjacent nodes. The average accuracy was very high 78.89%.

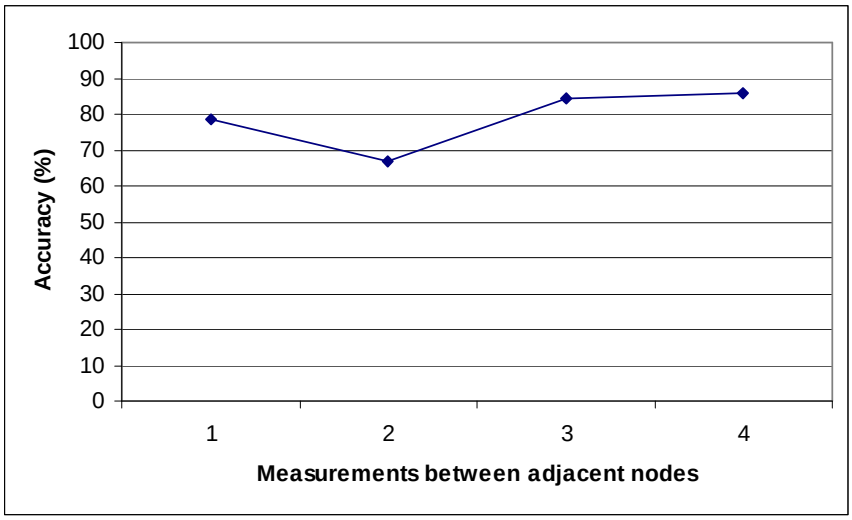

Fig 38 Measurements taken between adjacent forwarding nodes

5.3 Results comparison and evaluation of the system

After extensive testing in various configurations, with different number of beacons in outdoor and indoor environment we can conclude that the results of different measurements are somewhat coherent. Even in highly congested environment (indoor testing), the results of the tests seem to be acceptable if the routing nodes are far enough from each other.

As stated in the **5.1 Evaluation Goals** we wanted to identify which parts of the developed framework can be used for future development. The initial tests give valuable information on the performance of the system as a whole, but also lead to some conclusions concerning the efficiency of particular elements, like the routing protocol or the process of inferring the location information.

When talking about the evaluation, we need to keep in mind, that the present implementation is a mere prototype, a proof of concept rather than a ready solution. That is why the overall performance is not the most important issue.

When evaluating the system we need to consider the accuracy, which is the main goal to achieve. Another important points are latency and reliability of inferred information. In our opinion these are the most important factors.

5.3.1 Indoor tests' results comparison

The indoor and outdoor tests should not be directly compared. Due to huge differences in the propagation of the signal, the range of the motes and the "noisiness" of the environment a direct comparison would lead us to inaccurate and faulty conclusions.

When talking about indoor experiments it's easy to see that the overall performance of the system increased with the increasing distance between forwarding nodes. The reasons for that seem to be obvious. As it is easily visible from the pictures, the hall that the test was carried out in was very narrow and encouraged propagation of the signal which was reflected by the walls that were very close to the source of the signal. This is the only sensible explanation that is justified by the fact that moving the nodes more apart resulted in better experiment's results.

When comparing the tests' results in the graphs in Fig 27, Fig29 and Fig 31 that depict the measurements from three different experiments, carried out in the same environment but with different layout of ad-hoc network with distances between adjacent forwarding nodes of 3, 5 and 7 metres, it is easy to see that RSSI measurements and location calculation accuracy is increasing together with increasing distance between adjacent forwarding nodes. Overall accuracy values for these three experiments are 36,64%, 40,74% and 56,24%. The increase is meaningful and even though, the growing distance between adjacent motes reduces overall resolution of the system, the calculations are still more relevant to the actual location than those acquired during tests with small distances between forwarding motes.

Another issue worth mentioning is degraded performance of RSSI calculation if the motes that receive the signal are close to obstacles.

Looking at the graph in Fig 39 it is easy to notice reduced accuracy in estimating the location of the beacon around the mote number 2, 6 and 10. In a very close proximity to these three motes there were obstacles which we think influenced the results.

We have noticed similar behaviour in the test where the distance between the adjacent forwarding nodes was equal to 5 metres. We are not sure if this was only a coincidence as we haven't managed to conduct additional testing. However, the influence of obstacles on RSSI calculation is highly probable.

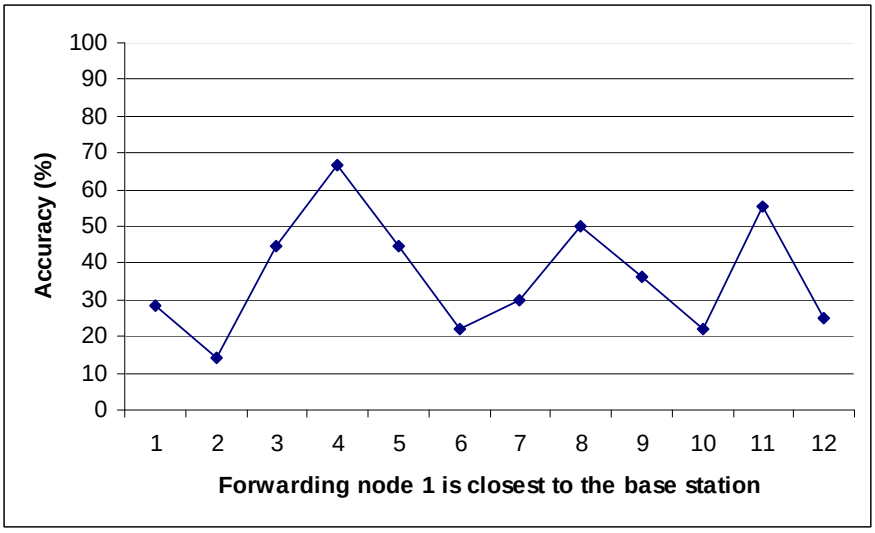

Fig 39 The results of the first static indoor test

The tests where the beacon was placed between two adjacent nodes seem as if they gave much better results. This is due to the fact that the test allowed for much greater error. We considered the result accurate when either of the IDs of the adjacent forwarding nodes was displayed by the base station's software. On the other hand the tests during which we were measuring the location when the beacon was directly next to a particular node did not allow for any error when measuring the accuracy. This is why the system seems to perform much worse in a situation when we measured the location of a beacon when it was placed directly next to one of the forwarding nodes. The results of the tests with beacon next to a particular node and between adjacent nodes should be taken into consideration together. According to this rule we managed to achieve:

- 46,82% accuracy – indoor test 3 metre distance between the nodes
- 54,76% accuracy – indoor test 5 metre distance between the nodes
- 71,29% accuracy – indoor test 7 metre distance between the nodes

The accuracy given above does concern only the IDs of the motes. This means that even though the accuracy of the system may seem the best when the distances between forwarding motes are 7 metres, we need to keep in mind that the error grows together with the distance between adjacent nodes.

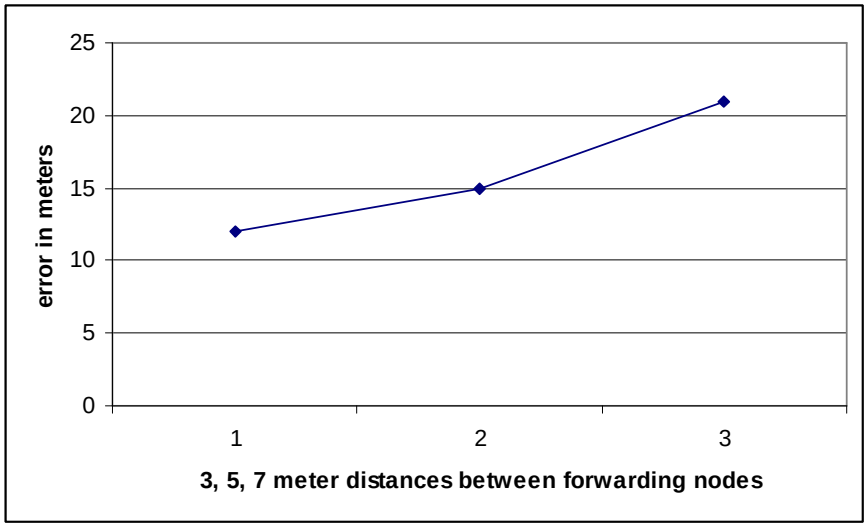

Fig 40 The distance between forwarding nodes vs. maximum error

The graph above shows the maximum error that we managed to log. When looking at this graph, one can say that the accuracy of inferring the location when the distance between the motes is 7 metres is not acceptable. It is important to say that among 13 different readings, there was only one that contained such a big error. One way of getting rid of such errors is to ignore odd readings that are not coherent with already acquired data.

In overall we were not surprised with the results of the indoor experiments. We were expecting large inconsistencies in the data gathered during the experiments. This is due to the environment that encourages the propagation of the signal. The results of these experiments were not coherent with the information about the range of Mica2 motes that we managed to gather before designing the system. According to this information, we predicted that the optimal distance between adjacent motes should not exceed 5 metres. It turned out however, that the results were better when we increased this distance to 7 metres.

We were quite surprised with poor resilience of the motes to the obstacles nearby that seemed to be distorting the signal and encouraging faulty RSSI readings.

We were very happy with the performance of the designed routing protocol which at this moment was the core part of the system. We were sceptical about it's performance in highly congested environment. It turned out that it managed large

numbers of messages very well. Either the tests with multiple beacons or the tests involving large numbers of forwarding nodes close to each other (12 motes – 3 metre distance) didn't seem to reduce it's performance. We haven't noticed any significant delay of the data arriving at the base station. The information was reliably displayed every 1,5 second at the base station.

5.3.2 Outdoor tests' results comparison

The results of the tests that we managed to carry out outdoors were much closer to our expectations than the indoors measurements. Both of the tests resulted in high accuracy of the data. As we predicted, the best performance was noted when forwarding nodes were 5 metres apart. This was consistent with the information acquired from the papers evaluating the range of Mica2 radio modules.

The data was also much more consistent and the maximum error had much lower value than the one observed in indoor tests. The graphs displaying the information are much smother than those showing the data for indoor tests results.

The overall performance of the system was very high:

- 83.57% accuracy – outdoor test 5 metre distance between the nodes
- 75,75% accuracy – outdoor test 7 metre distance between the nodes

The maximum error for both experiments was the same. It was equal to 2 motes, which results in 10 metre inaccuracy for 5 metre distances between the motes and 14 metre inaccuracy for 7 metre distances.

Again, we had no problems with the routing protocol which smoothly delivered all collected data.

The results of the experiments confirm our predictions concerning the layout of the ad-hoc network. Eventually it turned out that the distance of 5 metres between the motes seems to be optimal. Increasing it in outdoor environment resulted in decreased performance. What is more, the range of Mica2 radio modules decreases in the rain or fog, so the forwarding motes should not be placed further apart due to the possible loss of connectivity.

5.3.3 Overall system evaluation

The tests that we carried out helped us to answer important questions concerning the approach we have taken as well as the design and configuration of the system. We seem to have a definite answer concerning the routing protocol, which managed to handle the traffic much better than we expected and in our opinion is a good starting point for further development.

Slightly inconsistent indoor results should not be a big concern. The motes seem not to be resilient to the obstacles in the nearby and give faulty RSSI reading in case of objects present in close proximity. However, this should not create additional problems in the future.

The overall approach, considering the use of beacons for locating purposes seems to be a good temporary solution. It may help to develop the mechanisms that with slight changes may be implemented in future versions of the system.

The elements of the system that definitely need to be improved include either the base station software and the client software that will be installed in the car in the future. The major change that should be made to the base station software is the introduction of additional reasoning. If the base station was taking into consideration the direction of movement of a pedestrian and estimated speed; which might be calculated using multiple sensor readings; it might be possible to drop the faulty or odd readings.

The client's software, that receives broadcasts from the base station is very simple and needs further development as well.

Another issue that concerned us was 2 second latency in estimating the location of a walking pedestrian. The information displayed by the base station during a dynamic test were 2 seconds out of date. The speed of a person carrying a beacon during the test was a normal walking speed. The inaccuracy in predicted location if the person was running would be unacceptable. This might be improved by implementing more "intelligent" software for the base station.

We were not able to carry out the tests in bad weather conditions due to practical reasons. According to the information we managed to find, this might pose additional

problems, as the range of Xbow radio modules decreases in the rain and the fog. To avoid the loss of connectivity between the motes, the use of additional forwarding nodes acting only as routers and placed on a higher ground to reduce the effect of Fresnel zone might be considered. Further testing is required in order to identify the optimal distances between additional motes.

Chapter 6: Conclusion

6.1 General Conclusions

The solution presented in the previous parts of this dissertation is a fully functional prototype of the communication framework for Pedestrian Detection System. The core elements of the implementation that consists of the routing protocol deigned for the network of forwarding nodes participating in the process of gathering the information as well the process of measuring the RSSI signal performs well according to the results of our tests.

If similar equipment within ad-hoc network is used in the future, both the routing protocol and RSSI measurements may be used as a starting point for further development. Implemented communication framework should make it easier to deploy the network of sensors connected to the motes which will eliminate the necessity of using beacons for inferring the location of a particular person at a given time.

According to experiments that we carried out, the routing solution deployed for the system is fully scalable and is capable of handling large numbers of information, even in highly congested environment.

On the other hand many improvements in the base station's and client's software are needed. We outline the desired changes in Future Development section.

6.2 Objectives fulfilled

1. The initial design of the communication framework, based on ad-hoc network created between Mica2 motes as well as the development of base station's

software gathering the information acquired by the nodes in the network and multicasting these information over a wireless link.

2. The creation of basic mechanisms used for detection of pedestrians. These mechanisms are capable of inferring approximate location of the beacon with 10 metre accuracy.
3. Successful development of the routing protocol for the ad-hoc network; based on broadcasts and the assumption that the same packets can not be forwarded twice by the same node. This is the core element of the communication infrastructure.
4. The development of applications for the base station inferring the location from information gathered by particular nodes within the ad-hoc network.
5. The deployment of basic client application capable of receiving multicasted location information.
6. Initial testing and evaluation of the routing protocol, the performance and overall accuracy of the system and identification of elements of the system that may be used for further development.

6.3 Future Work

The number of possible improvements is vast and it is very hard to predict the direction of the evolution of the present system. We can outline some improvements which would strongly improve the overall performance of the system.

Adding additional functionality to the base station's software is a must. Present performance of inferring the position of a particular beacon is far to poor to be used. This requires additional work and testing.

The client's software is very primitive. Further work on this piece of software includes the deployment of the application for a handheld device and further testing of the solution in a wireless, remote configuration.

We predict that due to degradation of the range of Mica2 radio modules in bad weather, additional motes acting as routers might be necessary. These motes should be placed on a higher ground to increase the range of radio signal. Additional nodes

would act as a backup network for the motes embedded in the surface of the road. The final layout of additional nodes is a subject to further testing.

As the security should be a major concern when designing an Intelligent Transportation system, some security measures providing protection from threats outlined in the Design section of this dissertation should be introduced.

Future work should also include further development of mechanisms of detecting pedestrians. Finding a better detecting solution is beyond the scope of this paper. However, the use of some sort of sensors should be taken into consideration. This would eliminate the need of using the motes as beacons and thus make the system more effective and universal.

Appendices

Appendix A: Abbreviations

RSSI - Received Signal Strength Indicator

UDP - User Datagram Protocol

ITS - Intelligent Transportation System

DSRC - Dedicated Short Range Communications

PDS - Pedestrian Detection System

DSDV - Destination-Sequenced Distance-Vector routing protocol

PDU - Protocol Data Unit

AODV - Ad Hoc On-Demand Distance Vector Routing

RREQ - Route Request

RERR - Route Error

3G - third-generation

RVC - Road to Vehicle Communication

IVC - Vehicle Communication

ISM - Industrial, Scientific, Medical

FHSS - Frequency-hopping spread spectrum

DSSS - Direct-sequence spread spectrum

GHz - gigahertz

LLC - Logical Link Control

MAC - Media Access Control

IEEE - Institute of Electrical and Electronics Engineers

CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance

CSMA/CD - Carrier Sense Multiple Access with Collision Detection

Mbps - megabit per second

TTL - Time To Live

GPRS - General Packet Radio Service

UMTS - Universal Mobile Telecommunications System

GSM - Global System for Mobile Communications

kbps - kilobits per second

WiMAX - Worldwide Interoperability for Microwave Access

WiFi - Wireless Fidelity

TDM - Time-division multiplexing

IP - Internet Protocol

VoIP - Voice over Internet Protocol

TDMA - Time Division Multiple Access

dFCD - Decentralized Floating Car Data Services

Ultra-TDD - Ultra-time-division-duplex

CDMA - Code division multiple access

GPS - Global Positioning System

A/D converter - Analogue to digital converter

UART - Universal asynchronous receiver transmitter

TCP - Transmission Control Protocol

Bibliography

[1] Road Accident Facts 2002 (2003) Retrieved April 2006, from National Roads Authority Ireland, Road Safety - Downloadable Documentation Web site: http://www.nra.ie/PublicationsResources/DownloadableDocumentation/RoadSafety/file,684,en.PDF

[2] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, F. Chausse (2004). "Pedestrian Localization and Tracking System with Kalman Filtering". *IEEE Intelligent Vehicles Symposium, University of Parma, Italy, May 2004* Retrieved May 2006, from Dipartimento di Ingegneria dell'Informazione Universit`a di Parma: www.ce.unipr.it/people/broggi/publications/iv2004-tracking-clermont.pdf

[3] Harsh Nanda, Larry Davis (2002). "Probabilistic Template Based Pedestrian Detection in Infrared Videos". *IEEE Intelligent Vehicle Symposium, Versailles, France, June 18-20, 2002*. Retrieved May 2006, from Department of Computer Science, University of Maryland Web site:

[4] Elizabeth Royer, Chai-Keong Toh (1999). "A Review of Current Routing Protocols Ad Hoc Mobile Wireless Networks". Retrieved April 2006, from Electrical Engineering and Computer Science at Harvard University Web site: http://www.eecs.harvard.edu/~mdw/course/cs263/papers/royer-ieeepc99.pdf

www.cs.umd.edu/~nanda/Professional/Publications/IVS2002/iv2002.pdf

[5] Jijun Yin, Tamer ElBatt, Gavin Yeung, Bo Ryu, Stephen Habermas, Hariharan Krishnan, Timothy Talty (2004). "Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks". Retrieved February 2006, from UCLA Computer Science Department Web site: www.cs.ucla.edu/~gavin/pub/p52_Yin.pdf

[6] "Introduction to Wireless LAN and IEEE 802.11" Retrieved February 2006, from Tutorial-Reports Web site:
http://www.tutorial-reports.com/wireless/wlanwifi/index.php

[7] Eoin Bailey (2005). “An Implementation of a Parasitic Routing Algorithm”, Retrieved February 2006 from Computer Science Department The University of Dublin Trinity College Web site:
www.cs.tcd.ie/publications/tech-reports/reports.05/TCD-CS-2005-04.pdf

[8] William Lehr, Lee W. McKnight (2002) “Wireless Internet access: 3G vs. WiFi?” Retrieved February 2006, from:
itc.mit.edu/itel/Docs/2002/LehrMcKnight_WiFi_vs_3G.pdf

[9] Michael Meyer (1999). “TCP Performance over GPRS”. Retrieved February 2006 from: http://www.cs.helsinki.fi/u/gurtov/reiner/wcnc99.pdf

[10] Carl Eklund, Roger B. Marks, Kenneth L. Stanwood, Stanley Wang (2004). “IEEE Standard 802.16: A Technical Overview of the WirelessMAN™ Air Interface for Broadband Wireless Access”. Retrieved March 2006, from: http://www.ieee802.org/16/docs/02/

[11] Michael Meincke, Peter Tondl, María Dolores, Pérez Guirao, Klaus Jobmann. (2002). “Wireless Adhoc Networks for Inter-Vehicle Communication”. Retrieved May 2006, from Universität Hannover – Fakultät für Elektrotechnik und Informatik Institut für Kommunikationstechnik Web site:
http://www.ant.unihannover.de/Forschung/ Public/Kn/2002/MTP2002.pdf

[12] Dr. Walter J. Franz, Dr. Hannes Hartenstein, Bernd Bochow (2001) “Internet on the Road via Inter-Vehicle Communications”. Retrieved March 2006, from Technische Universität Hamburg-Harburg Web site: http://www.et2.tu-harburg.de/fleetnet/pdf/GI_WShop_FleetNet.pdf

[13] - Walter Franz, Christian Maihöfer (2002) “Geographical Addressing and Forwarding in FleetNet”. Retrieved May 2006, from FleetNet Web site: http://www.et2.tu-harburg.de/fleetnet/pdf/

[14] Daniel V. McGehee Elizabeth N. Mazzae G.H. Scott Baldwin (2000). “Driver Reaction Time in Crash Avoidance Research: Validation of a Driving Simulator Study on a Test Track” Retrieved April 2006, from US National Highway Traffic Safety Administration: www-nrd.nhtsa.dot.gov/vrtc/ca/capubs/IEA2000_ABS51.pdf

[15] Samir Goel, Tomasz Imielinski, and Kaan Ozbay (2004). „Ascertaining Viability of WiFi based Vehicle-to-Vehicle Network for Traffic Information

Dissemination" *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on 3-6 Oct. 2004* Retrieved February 2006, from: http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1399058

[16] Mica2 series technical specs. Retrieved February 2006, from Xbow Web site: http://www.xbow.com/Products/productsdetails.aspx?sid=72

[17] MIB 510 technical specs. Retrieved February 2006, from Xbow Web site: http://www.xbow.com/Products/productsdetails.aspx?sid=79

[18] Mission Statement, Retrieved March 2006, from TinyOS Web site: http://www.tinyos.net/special/mission

[19] Anastasi, A. Falchi, A. Passarella, M. Conti, E. Gregori (2004). "Performance Measurements of Motes Sensor Networks". Retrieved February 2006, from University of Pisa, Facoltà di Ingegneria Web site: http://www2.ing.unipi.it/~o783499/research/docs/MSWiM04.pdf

[20] "CC1000 Single Chip Very Low Power RF Transceiver Data Sheet v.2.1", Retrieved February 2006, from Texas Instruments Web site: http://focus.ti.com/lit/ds/symlink/cc1000.pdf

[21] TOS message structure, Retrieved February 2006, from: http://mail.millennium.berkeley.edu/pipermail/tinyos-contrib-commits/2005February/001331.html

[22] TinyOs Tutorial, Lesson 6: Displaying Data on a PC, Retrieved February 2006, from: http://www.tinyos.net/tinyos-1.x/doc/tutorial/lesson6.html