

Research Article

Heterogeneous Wireless Sensor Network for Transportation System Applications

K. Selvarajah,¹ C. Shooter,² L. Liotti,³ and A. Tully¹

¹ School of Computing Science, Newcastle University, Newcastle NE1 7RU, UK

² TRW Conekt, Stratford Road, Solihull B90 4AX, UK

³ Process Research, Centro Ricerche FIAT, Strada Torino, 50, 10043 Orbassano (TO), Italy

Correspondence should be addressed to K. Selvarajah, k.selvarajah@ncl.ac.uk

Received 6 August 2010; Accepted 2 March 2011

Academic Editor: Syed R. Rizvi

Copyright © 2011 K. Selvarajah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The important innovations in wireless and digital electronics will support many applications in the areas of safety, environmental and emissions control, driving assistance, diagnostics, and maintenance in the transport domain. The last few years have seen the emergence of many new technologies that can potentially have major impacts on transportation systems. One of these technologies is Wireless Sensor Networks. A wireless sensor device is typically composed of a processing unit, memory, and a radio chip which allows it to communicate wirelessly with other devices within range. The Embedded Middleware in Mobility Applications (EMMA) project delivers a middleware that aims to facilitate the interaction between sensing technologies in transportation systems. This paper outlines our experience in the EMMA project and provides an illustration of the important role that wireless sensor technology can play in future transportation system. The paper discusses our experience of using heterogeneous sensors to develop transportation system applications in the EMMA project and focuses on how cooperation between vehicle and infrastructure can be addressed. It also presents encouraging results obtained from the experiments in investigating the feasibility of utilising wireless sensor in vehicle and vehicle-to-infrastructure communication in real transportation applications.

1. Introduction

A recent study by the UK Government's Office of Science and Innovation, which examined how future intelligent infrastructure would evolve to support transportation over the next 50 years, looked at a range of new technologies, systems, and services that may emerge over that period [1, 2]. One key class of technology that was identified as having a significant role in delivering future intelligence to the transport sector is Wireless Sensor Networks (WSN) and in particular the fusion of fixed and mobile networks to help deliver a safe, sustainable, and robust future transport system based on the better collection of data, its processing and dissemination, and the intelligent use of the data in a fully connected environment. As future intelligent infrastructure will bring together and connect individuals, vehicles and infrastructure through wireless communications, it is critical that robust communication protocols are developed.

Mobile wireless ad-hoc networks (MANETs) are self-organising networks where nodes exchange data without the

need for an underlying infrastructure [3]. MANETs have attracted extraordinary attention from the research community in recent years including in real transport applications. In the road transport domain, schemes which are fully infrastructureless and those which use a combination of fixed (infrastructure) devices and mobile devices fitted to vehicles and other moving objects are of significant interest to the transport community as they have the potential to deliver a "connected environment" where individuals, vehicles, and infrastructure can coexist and cooperate, thus delivering more knowledge about the transport environment, the state of the network, and who indeed is travelling or wishes to travel [4]. This may offer benefits in terms of real-time management, optimisation of transport systems, intelligent design, and the use of such systems for innovative road charging and possibly carbon trading schemes as well as through the CVHS (Cooperative Vehicle and Highway Systems) for safety and control applications. Within the vehicle, the devices may provide wireless connection to various Information and Communications Technologies (ICT)

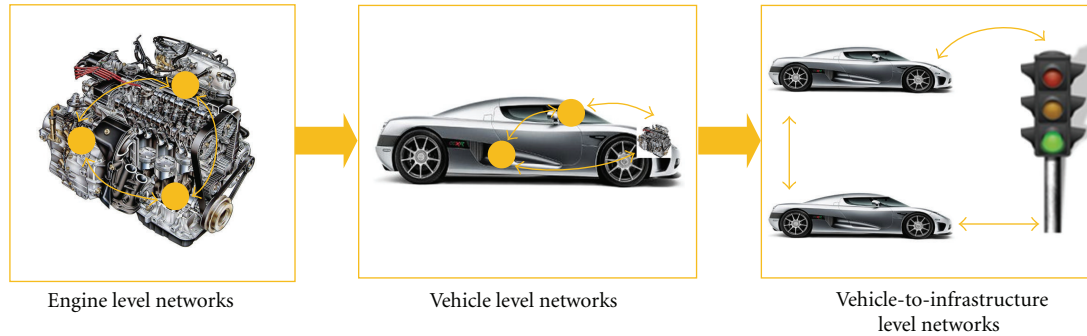


FIGURE 1: EMMA hierarchical network.

components in the vehicle and connect with sensors and other devices within the engine management system [5]. Advances in wireless sensor networking techniques which offer tiny, low power and MEMS (Micro-Electro Mechanical Systems) integrated devices for sensing and networking will exploit the possibility of vehicle-to-vehicle and vehicle-to-infrastructure communications [6].

In this paper, wireless sensor network applications in the transport systems and using middleware to integrate heterogeneous Wireless Cooperative Objects (WICOs) are discussed. Section 2 describes the EMMA project and its hierarchical approach and communication technologies. An overview of wireless sensor networks middleware and components of the EMMA middleware is presented in Section 3. Section 4 describes the different hardware platforms which are used as wireless cooperating objects in the prototype applications. Three applications for each hierarchical level and an inter-hierarchical level application are given in Section 5. Section 5 also presents encouraging results obtained from the experiments in investigating the feasibility of utilising EMMA middleware in real transport system applications. Conclusions are then presented in Section 6.

2. The EMMA Project

The EMMA project (Embedded Middleware in Mobility Applications project) is partly funded by the European Commission under the Information Society Technologies (IST) Priority of the 6th Framework Programme. The EMMA project was committed to deliver a middleware platform and a development environment which facilitates the design and implementation of embedded software for cooperative sensing objects [7, 8]. The EMMA network architecture (Figure 1) can be considered at three levels: within an engine level, at a vehicle level, and at the supra-vehicle level. Recently, many wireless sensor network applications have been developed for a variety of applications including transport monitoring and control. However, there are still numerous challenges to be overcome if wireless sensor devices are to communicate with each other in an intelligent, cost-effective, and reliable way.

EMMA communication networks at the supra-vehicle level can be considered as mobile wireless sensor networks

while vehicle level networks and engine level networks can be considered as static wireless sensor networks. The current wireless sensor networks employ conventional technologies to interact with other devices in the network, and many companies and organizations are developing various wireless communication interface and protocols for sensors. Bluetooth is currently the most widely used automotive wireless technology for in-vehicle communication and Wi-Fi is used for vehicle to vehicle communication by several pilot research projects such as the Car2Car consortium [9]. ZigBee technology is able to provide the interconnection of low power wireless sensors within vehicles and vehicle to infrastructure. The ZigBee standard has evolved since its original release in 2004 and it is a low cost low power wireless networking standard for sensors and control devices. ZigBee provides network speed of up to 250 kbps and is expected to be largely used in typical wireless sensor network applications where high data rates are not required [10, 11].

The EMMA project needed to discover which communication technologies are more suitable and how the networks are formed by WICOs from different levels. ZigBee, Bluetooth and Wi-Fi have been designed for short-range wireless applications with low power solutions and could be used at the EMMA infrastructure level. ZigBee can accommodate larger numbers of devices than Bluetooth. On the other hand, Bluetooth offers high bandwidth with relatively high throughput. EMMA network applications do not require high data rate communication technology as it is based on data exchange. ZigBee provides 250 kbps data rate and is expected to be enough for the EMMA sensor network applications. Notably, ZigBee uses low overhead data transmission and requires low system resources which are vitally important factors for embedded wireless sensor networks. Also mesh networking features in ZigBee technology allow devices to extend coverage and optimize radio resources. These features show that ZigBee is a suitable communication technology for the EMMA project applications.

3. Middleware for Wireless Sensor Networks

The term middleware refers to the software layer between the operating system and the applications. A middleware layer seeks primarily to hide the underlying network environment complexity by insulating applications from explicit

protocol handling, disjoint memories, data replication, network faults, and parallelism. Further middleware masks the heterogeneity of computer architectures, operating systems, and communication technologies to facilitate application programming and management [12]. The design and development of a successful middleware should address many challenges in WSN such as scarcity of resources, mobility, heterogeneity, data aggregation, quality of services, and security. Several middleware systems have been proposed for WSN but each addresses a different part of the problem space. Notable middleware for sensor networks are Impala, Mate, TinyDB, TinyCubus, TinyLime, and MiLAN [13]. Most of these middleware are built on top of TinyOS [14] which is an open source operating system mainly designed for wireless sensor networks. The middleware can be classified as service-centric middleware and data-centric middleware. Service centric middleware is driven by commands while data centric middleware is driven by data.

Service-centric middleware is described as a well-defined and self-contained function that does not depend on the context or the state of other services. Such a service is executed by explicitly calling it. After the completion of the service, a response is returned. This type of middleware is the principally used paradigm in traditional distributed systems, either with a procedural abstraction or based on object-orientation. Data-centric middleware is mostly concerned with the communication of data and provides a small general purpose API to send and receive data. There is no client-server relationship but there is a distinction between data providers and data consumers. The data-centric approach is mainly followed in the area of sensor networks where the naming and type of data play a more important role than the specific device responsible for its processing.

The EMMA Embedded Middleware Platform (EM2P) is designed to support a range of applications running on different WICOs. EM2P is designed in a modular fashion. The communication adapters form the interface between the communication module and the actual hardware drivers. The communication module has a generic part and two specialised parts for message and data-centric communication. The security add-on is configurable via the middleware API, but is actually used in the communication module. The same applies for the data connector. Installation and the configuration and monitoring module use message communication and are, therefore, built on top of the message communication. A general communication abstraction is used by the synchronisation module. The main components of the EM2P is shown in Figure 2.

The middleware abstracts from the underlying communication technology by providing a high-level addressing mechanism and the communication functions do not imply a specific communication technology. The middleware converts the local representation of a data value to its network representation and vice versa when sending or receiving data. Messages can be sent directly to a specific WICO by knowing only its EMMA WICO address. An application can register for the reception of messages. A call back function is called when a message is received. The content of a message is completely controlled by the application and

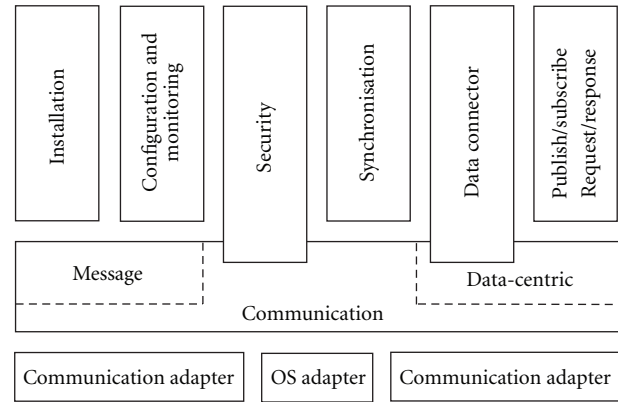


FIGURE 2: Components of EMMA Middleware.

no data conversion is done by the middleware. Therefore, the application has to assure that the receiver understands the message contents. EM2P uses publish/subscribe communication, request/response communication, and data connector functionalities.

4. EMMA Wireless Cooperative Objects

The following sections explain three different platforms which are used in EMMA project prototype applications: Commercially available Crossbow MicaZ and TelosB and an off-the-shelf Xilinx ML403 FPGA board. C-based multi-threaded NanoQplus [15] operating system is used in MicaZ and TelosB motes while a Linux-based Qplus [15] operating system is used in Xilinx ML403 FPGA. These devices with sensors, actuators, and related software are called WIREless Cooperating Objects (WICOs) which may be heterogeneous, but nevertheless able to cooperate together to achieve specific goals [8].

4.1. Supra-Vehicle Level WICO. Smartdust (or mote) is a new concept for wireless sensor networks which offers tiny, low-power, and MEMS integrated devices for sensing and networking. It is interesting for the low-power sensing technologies and also the low-power communication and networking capability which it has demonstrated. Fundamentally, it provides a convenient and economic means of gathering and disseminating environmental and other useful information in the transport domain. The existence of a ZigBee-based networking capability between the motes and other devices will benefit many applications in the transportation domain. The motes have sensors attached to them to monitor the physical environment in some way. These sensors can be built directly onto the mote or can come as daughter boards which can be connected to the motes main mother board. Initial studies suggest environmental monitoring, vehicle-to-vehicle, vehicle-to-infrastructure, and infrastructure to infrastructure applications may exist for motes in the transportation domain. The vital application of the devices is beginning to be tested in the road vehicle environment. Even though a range of mote platforms are



FIGURE 3: MicaZ WICO.

available in the market, Crossbow MicaZ [16] family motes had been chosen for EMMA project as it features sensing and networking capabilities with low-power consumption using ZigBee as communication protocol. Figure 3 shows a Crossbow MicaZ mote.

The MicaZ is a family of the Crossbow Mica motes where the radio transceiver uses the Chipcon CC2420 IEEE 802.15.4 (ZigBee) compliant chipset. This allows the MicaZ to communicate with other ZigBee compliant equipment. The software stack includes a MicaZ mote-specific layer with ZigBee support and platform device drivers, as well as a network layer for topology establishment and single/multihop routing features. It is mainly used for research and development of low power wireless sensor network applications. The MicaZ mote platform is built around the Atmel AtMega128L processor which is capable of running at 7.37 MHz. The MicaZ motes have 128 Kbytes of program memory, 512 Kbytes of flash data logger memory, and 4 Kbytes of SRAM. Power is provided by two AA batteries, and the devices have a battery life of roughly one year depending on the application (very low duty cycle assumed). Sensor boards can be attached through a surface mount 51 pin connector, Inter-IC (I2C), Digital Input Output (DIO), Universal Asynchronous Receiver Transmitter (UART), and a multiplexed address/data bus.

4.2. Engine Level WICO. In the Engine level application, the Crossbow TelosB [16] mote platform was chosen for the EMMA project prototype applications. Compared to the MicaZ mote, the TelosB mote has higher processing power which is required to implement with a multitasking approach applications: to run several threads of the middleware, the acquisition task, and keeping low latencies in the communication among the Engine WICOs. The high ADC resolution is necessary to satisfy one of the needs of the engine application that is the acquisition of an accurate analogue signal. Less important but good features are the presence of a USB connection, the on board sensors, and LEDs for demonstration and development purposes.

The Crossbow TelosB (Figure 4) mote is a commercially available mote platform with Chipcon CC2420 IEEE



FIGURE 4: TelosB WICO.

802.15.4 (ZigBee) compliant chipset with integrated onboard antenna. This allows the TelosB mote to communicate with other ZigBee compliant equipment such as MicaZ. It is also mainly used for research and development of low-power wireless sensor network applications. The TelosB mote platform is built around the TI MSP430 (a 16-bit microcontroller) which is capable of running at 8 MHz. The TelosB mote has 48 Kbytes of program memory, 1024 Kbytes of flash data logger memory, and 10 Kbytes of SRAM. Power is provided by two AA batteries, and the devices have a battery life of roughly 1 year depending on the application (very low duty cycle assumed). The TelosB mote has 12 bit ADC resolution, while the MicaZ mote has 10 bit resolution. Sensor boards can be attached through Inter-IC (I2C), Digital Input Output (DIO), Universal Asynchronous Receiver Transmitter (UART), and SPI.

4.3. Vehicle Level WICO. At the vehicle level, there is a need for introducing wireless communication to existing sensing technologies. In addition, it may be necessary to increase the processing power and memory space in order to ensure the EMMA middleware runs seamlessly without compromising the performance of each sensor. It is important to keep the low-power communications available in the other hardware alternatives used but it may be necessary for the units themselves to be capable of running much more complex algorithms. The hardware chosen for EMMA project for this level of WICO, therefore, reflects that extra processing power needed. The vehicle level WICO consists of two elements. Firstly, an off-the-shelf Xilinx ML403 FPGA [17] board is used as the foundation of the system. This FPGA contains a powerful Power PC microprocessor. Secondly, the functionality in this board is extended using a custom, built daughter board. This daughter board contains a number of different hardware devices required by the project including a 12 V automotive power supply, a CAN [18] port for interfacing to automotive ECUs and 2 further RS232 ports which are used to send and receive data over ZigBee and from the other devices as appropriate (e.g., GPS). Figure 5 shows Xilinx ML403 FPGA with TRW Conekt interface board.

5. Prototype Applications

This section provides four different prototype applications, their implementation, and related experiments. The applications are developed here not only to evaluate the overall

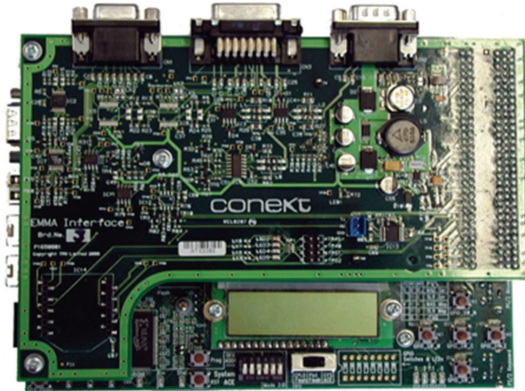


FIGURE 5: Xilinx processor board with TRW Conekt interface board mounted on top.

results of the project, but also to demonstrate the validity of the EMMA approach and its potential applications of heterogeneous wireless networks in transport systems.

5.1. Engine Level Application. The application proposed in the EMMA project is a solution for new engine control architecture (Figure 6), characterised by the integration of new sensors (in-cylinder pressure, oil pressure, and valve lift, not available on current engines) without redesigning the ECU engine. The engine network (Figure 7), wirelessly connected with the ZigBee technology, is so composed:

- (i) 4 Cylinder WICOs, sampling the two sensors (a pressure sensor and a valve lift sensor) on each cylinder,
- (ii) oil pressure WICO, sampling by a sensor for the measurement of the oil pressure in the oil delivery head,
- (iii) ECU WICO, composed by a wireless node connected to the ECU (Electronic Control Unit).

The role of the four Cylinder WICOs is to sample the two analogue channels (connected to the valve lift and the in-cylinder pressure sensors) and calculate the maximum value for the first and the integral over a whole engine cycle for the second. The oil pressure sensor is responsible for sampling the oil pressure sensor. Upon a request from the ECU (simulated by a LabVIEW software on a Laptop), the ECU WICO queries the other engine WICOs, collects the received messages, calculates their latency and validity, and returns the data to the ECU.

5.1.1. Implementation. The Engine WICO consists of two main components: a TelosB mote and hardware adaptation module, necessary to properly interface the connection available on the board to the required conditioning electronics required by the engine sensors. The application has been evaluated by an ad-hoc test bench, where the values acquired from the real sensors are reproduced on the analogue outputs of an acquisition board, reading them from a set of measurement previously acquired from a real Multijet

engine. The ECU is implemented by the software running on a Laptop.

All the WICO applications have been programmed by the EM2P (EMMA middleware) functionalities. Several set of experiments have been carried out for data centric paradigm (request/response).

5.1.2. Experimental Results. This engine application has been validated in three different scenarios: the whole test bench has been tested in the laboratory environment and in the environmental chamber, while the engine nodes (without the acquisition board) have been tested directly on a real engine. A set of tests have been performed for the three different environments based on EMMA request/response mechanism (*data-centric*), and for several values of RPM (from 1000 rpm to 6000 rpm). For each test, a set of log files (registering latencies, packet loss, and other WSN-related data) have been collected for offline analysis of the application performance.

Laboratory Environment. The laboratory tests were carried out on a test bench using an engine simulator for the six different rpm values (1000 to 6000 rpm, step 1000). Table 1 summarises the results of each log file generated from each test run. The average and standard deviation of each cylinder's latency has been calculated using only complete packets, that is, where all cylinder WICOs returned a packet flag of 0.

Figure 8 shows that the average latency of all Cylinder WICOs increases slightly in-line with the rpm, whilst the standard deviations are consistent throughout all rpm. These results suggest that it is also possible to make a *data-centric* measurement of the engine sensors for a single engine cycle by simply performing a request with the necessary advance. Packet loss is under 2% for all rpm values, which demonstrates good communication stability for the *data-centric* paradigm.

Engine Environment. The engine environment tests were conducted as before, using a petrol engine to measure the influence of electromagnetic noise and the presence of metal objects in close proximity to the WICOs, in order to model the effects of "real world" conditions. The tests were carried out using an engine simulator for four different rpm values (1000, 2000, 3000, and 6000 rpm), all with the engine switched on. Table 2 summarises the results of each log file generated from each test run. The average and standard deviation of each cylinder's latency has been calculated using only complete packets, that is, where all cylinder WICOs returned a packet flag of 0.

Figure 9 shows that the average WICO latency increases in-line with the rpm for Cylinder 1 WICO, Cylinder 3 WICO, and Cylinder 4 WICO. Although the average latency of Cylinder 2 WICO actually decreases slightly at rpm value 2000 it increases for higher rpm values. The standard deviations of all Cylinder WICO latencies are consistent at all rpm. This further supports the notion that for *data-centric* communication, it is possible to make a measurement of the engine sensors for a single engine cycle by simply performing a request with the necessary advance. Packet loss

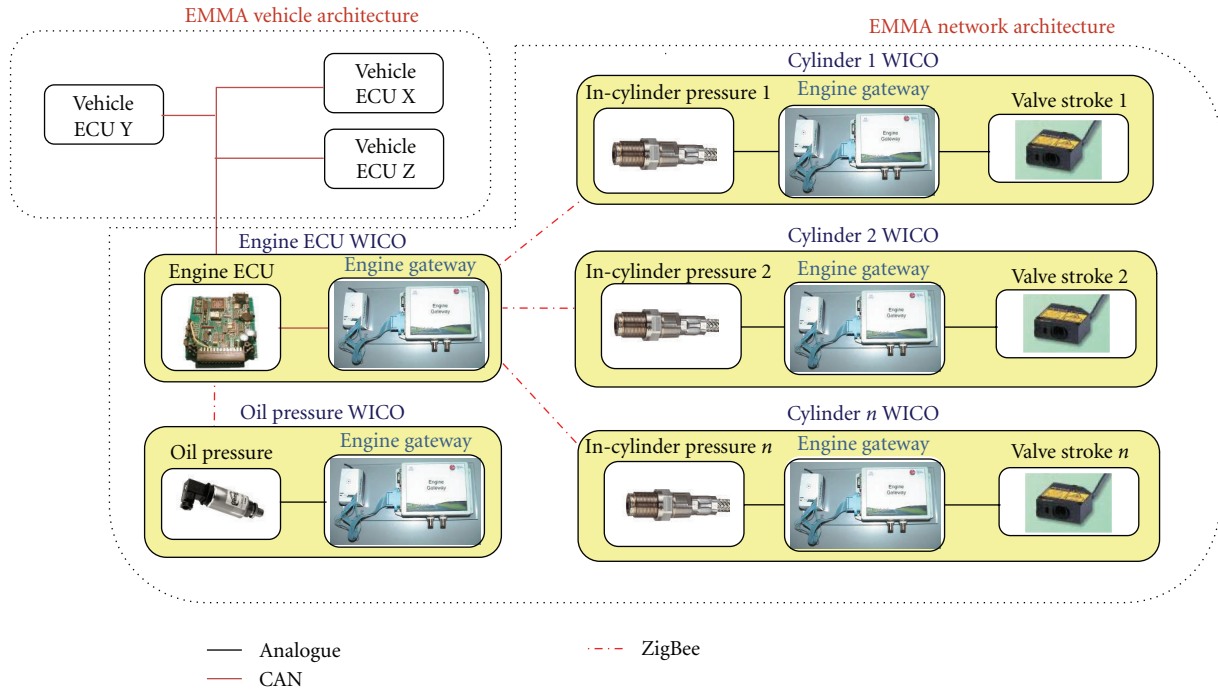


FIGURE 6: Engine level application diagram.

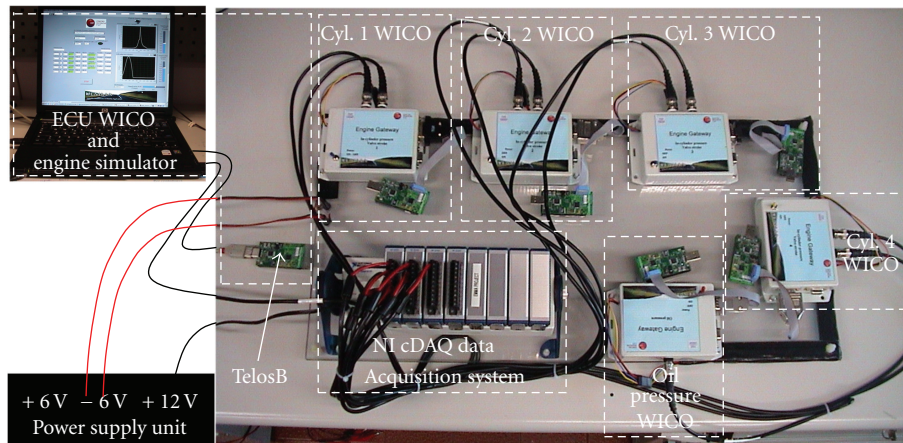


FIGURE 7: Engine WICOs experimental setup: acquisition board, 6 WICOs and ECU implemented on a Laptop.

TABLE 1: Summary of test results for laboratory environment.

rpm	Number of enquiries	Lost engine data* (n, %)	Cylinder WICO latency average (ms), standard deviation (ms)				Serial latency average (ms), standard deviation (ms)
			Cyl. WICO1	Cyl. WICO2	Cyl. WICO3	Cyl. WICO4	
1000	5717	88, 1.5	27, 3	4, 2	20, 3	16, 3	119, 7
2000	2603	12, 0.5	33, 3	11, 2	27, 3	23, 4	125, 6
3000	2952	36, 1.2	35, 2	13, 2	29, 3	25, 3	128, 3
4000	7445	72, 1.0	37, 2	15, 2	30, 2	26, 3	128, 6
5000	2631	24, 0.9	37, 2	15, 2	31, 2	27, 3	128, 3
6000	2024	12, 0.6	38, 3	16, 2	31, 3	27, 3	130, 4

*Excluding lost packets from Oil WICO.

TABLE 2: Summary of test results for engine environment.

rpm	Number of enquiries	Lost engine data* (n, %)	Cylinder WICO latency average (ms), standard deviation (ms)				Serial latency average (ms), standard deviation (ms)
			Cyl. WICO1	Cyl. WICO2	Cyl. WICO3	Cyl. WICO4	
1000	1810	16, 0.9	10, 2	26, 3	3, 2	19, 2	124, 5
2000	3731	30, 0.8	11, 2	19, 3	33, 3	28, 3	132, 6
3000	1644	12, 0.7	13, 2	21, 2	35, 2	31, 3	132, 5
4000	—	—	—	—	—	—	—
5000	—	—	—	—	—	—	—
6000	1967	24, 1.2	15, 3	23, 2	37, 3	33, 3	133, 3

*Excluding lost packets from Oil WICO.

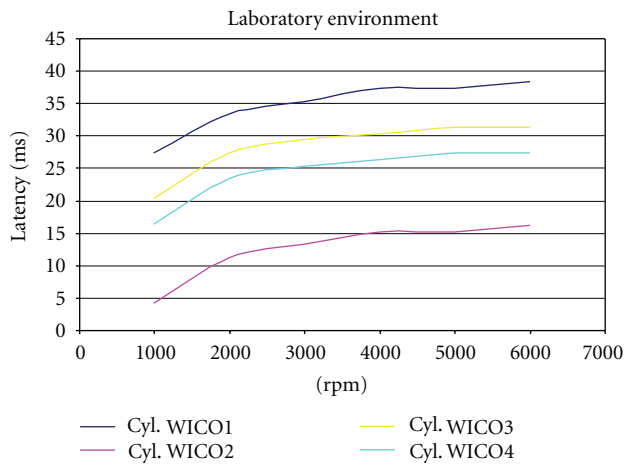


FIGURE 8: Cylinder WICOs latencies in laboratory environment.

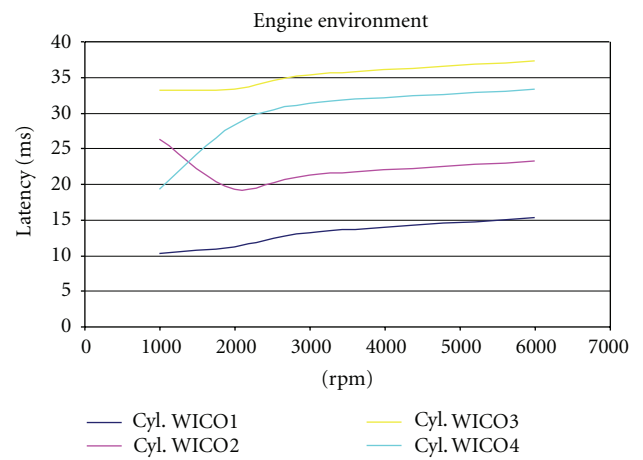


FIGURE 9: Cylinder WICOs latencies in engine environment.

in the engine environment is low, around 1% for all rpm, demonstrating good communication stability for the *data-centric* paradigm.

Environmental Chamber. Tests were carried out using an environmental chamber to control temperature and humidity conditions. The Cylinder 1 and Cylinder 2 WICOs were placed inside the chamber and the internal temperature varied for each test, whilst the remaining WICOs were placed outside the chamber during each test. Test runs were conducted using constant temperatures of -10 , 10 , 30 , 50 , and 80°C but unlike the laboratory and engine tests, these tests were only undertaken at two different rpm, 1000 and 3000, so the main variable investigated here was the effect on the WICO performance of the temperature inside the chamber. Tables 3 and 4 summarise these tests.

The results of the *data-centric* environmental chamber tests indicate that there is an effect of higher temperatures on the performance of the WICOs. To confirm this finding, the constant temperature tests were followed by a further variable temperature test. This applied a positive temperature ramp to the WICOs in the environmental chamber from -30 to 90°C using an rpm of 1000.

Table 5 presents the results of the variable temperature test which further illustrates the effects of temperature on the WICO latency values. The WICOs inside the chamber (Cylinder 1 and Cylinder 2) have higher standard deviations than those outside and overall packet loss is also slightly higher than in the other tests, at 3%. Figure 10 illustrates how the latencies of each WICO changed as the temperature ramp was applied, and clearly demonstrates the effect of increasing temperature on the WICO latency.

For the laboratory and engine environments, the latency standard deviations of all Cylinder WICOs were remarkably consistent, which demonstrates robust data communication stability across both environments for the *data-centric* paradigms. For the Environmental Chamber tests, an increase in the average latency of the Cylinder WICOs within the chamber was observed along with an increase in the standard deviations, which indicates a slight loss in communication stability for the *data-centric* paradigms.

The results of the environmental chamber tests clearly illustrate the impact of higher temperatures on WICO latency performance and the packet loss rate. The impact on latency performance is particularly noticeable in the Environmental Chamber tests where the differences in individual

TABLE 3: Summary of test results for environmental chamber at 1000 rpm.

Temp °C	Number of enquiries	Lost engine data* (n, %)	Cylinder WICO latency average(ms), standard deviation (ms)				Serial latency average (ms), standard deviation (ms)
			Cyl. WICO1	Cyl. WICO2	Cyl. WICO3	Cyl. WICO4	
-10	2295	44, 1.9	28, 3	5, 2	22, 3	17, 3	123, 6
10	1529	8, 0.5	9, 3	16, 3	26, 2	5, 2	119, 6
30	1868	16, 0.9	19, 2	30, 5	27, 3	6, 5	122, 7
50	1359	8, 0.6	2, 4	1, 7	8, 6	17, 5	92, 5
80	5652	68, 1.2	45, 4	39, 4	13, 5	6, 5	123, 6

*Excluding lost packets from Oil WICO.

TABLE 4: Summary of test results for environmental chamber at 3000 rpm.

Temp °C	Number of enquiries	Lost engine data* (n, %)	Cylinder WICO latency average (ms), standard deviation (ms)				Serial latency average (ms), standard deviation (ms)
			Cyl. WICO1	Cyl. WICO2	Cyl. WICO3	Cyl. WICO4	
-10	2354	48, 2.0	36, 2	14, 2	30, 2	25, 3	128, 3
10	1608	28, 1.7	19, 2	26, 3	35, 2	14, 2	127, 2
30	2147	24, 1.1	23, 2	32, 4	35, 3	17, 3	127, 2
50	1350	8, 0.6	35, 3	34, 7	26, 6	17, 5	122, 6
80	2236	23, 1.0	28, 3	40, 4	36, 3	14, 3	127, 4

*Excluding lost packets from Oil WICO.

TABLE 5: Summary of test results for temperature ramp within the environmental chamber.

Ramp	Number of enquiries	Lost engine data* (n, %)	WICO latency average (ms), standard deviation (ms)				Serial latency average (ms), standard deviation (ms)
			Cyl. WICO1	Cyl. WICO2	Cyl. WICO3	Cyl. WICO4	
+ve	2225	67, 3.0	34, 15	32, 13	7, 5	14, 5	122, 8

*Excluding lost packets from Oil WICO.

WICO latencies are primarily due to the experimental setup of the tests performed.

During these tests, two Cylinder WICOs were placed in the Environmental Chamber (Cylinder 1 and 2 WICOs) while the others (ECU WICO and Cylinder 3 and 4 WICOs) were placed outside. As the temperature increases, the time to perform the channel sampling and the calculation increases for the TelosB in the Environmental Chamber. For this reason, both the operations are completed first for the Cylinder WICOs 3 and 4, and then for the Cylinder WICOs 1 and 2 at higher temperatures, not in the serial order as expected.

As an example, where the bit rate of the ZigBee channel is around 40 kbps, if the answer length of each Cylinder WICO is around 800 bits, each answer will keep the ZigBee channel busy for at least 800 b/40000 bps (=20 ms). With the overhead of the ZigBee and the overhead introduced by the EMMA middleware, a difference between the latency

values of two contiguous Cylinder WICOs of about 40 ms is understandable. This explains why the latency increases for the Cylinder WICOs 1 and 2 in the Environmental Chamber, and therefore the reason why they are the last in the answer message queue.

The reasons for the increase in the loss rate can be attributed to the fact that all electronic devices have an operating point at which they can operate normally. As the temperature increases, I/V (current-voltage) characteristic of a device changes and the behaviour of the device can be different from what would be expected under "normal" conditions. For the WICOs, this change in I/V characteristics means that there is a possibility that a ZigBee transmitter chip behaves erroneously, which in turn produces corrupted data at the chip level. If an error does occur, repeated retransmission occurs on a chip as well as at the MAC level.

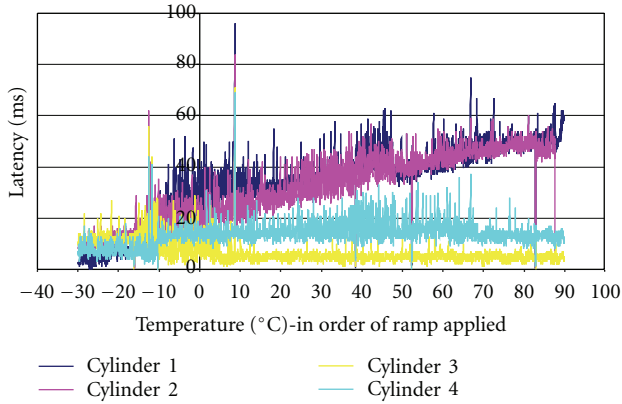


FIGURE 10: Cylinder WICO Latencies, positive temperature ramp applied.

The experimental results of all tests performed on the three scenarios have highlighted some key findings and issues.

- (1) The latencies are quite stable with the RPM, but they increase when the temperature reaches 50–60 degrees: this suggests that the TelosB mote requires further hardware design development for an automotive engine level application.
- (2) For the message centric version of the application, a loss packet rate lower than the data centric version has been observed: this suggest that in the two paradigms there is a different call-back implementation, or in the thread management in the OS.
- (3) For high values of RPM, ZigBee was unable to manage synchronously the connection between the ECU WICO and the 4 Cylinder WICOs: in fact the amount of data transmitted by each Cylinder WICO keep the RF channel busy for a number of millisecond comparable with the engine period, and this does not allow the ECU WICO to collect the data from the Cylinder WICO at the same time.

5.2. Vehicle Level Application. Figure 11 shows the example application that was developed to test the EMMA system. The overall purpose of the vehicle WICO was to provide the absolute position of targets being tracked by the radar. This was achieved by combining the absolute position of the vehicle (based on GPS & vehicle dynamics data) and the relative position of the detected target (using automotive ACC radar).

NMEA 0183 format data was used to transmit the GPS and vehicle data to the Radar WICO within the system. This standardised format was selected to ensure maximum interoperability of the individual WICOs in future setups.

5.2.1. Implementation. All implementations on the Xilinx ML403 FPGA board follow the generic architecture layout described below in Figure 12. Creation of the applications was carried out utilising the Qplus [15] operating system, which is based on Linux and the EMMA middleware.

TABLE 6: Summary of latency results for vehicle-level request/response.

Metric (in ms)	Run 1	Run 2	Run 3
Minimum	66	58	54
Maximum	475	436	452
Average	267	267	273
Standard deviation	400	483	566

GPS WICO. The interface to the Radar WICO was implemented using the publish/subscribe EM2P functionality to send the required GPS NMEA sentences to the main Radar WICO when they have been correctly received from the GPS unit. During development of the host vehicle tracking algorithm, it was discovered that only the GPGLA NMEA sentence was required for the application so all other sentences were filtered out by the GPS WICO.

Vehicle Dynamics WICO. The interface to the Radar WICO was implemented using the request/response EM2P functionality to send the latest vehicle dynamics data to the main Radar WICO when requested. The WICO buffers the received data from the individual sensors and sends the latest full update when requested. The NMEA sentence received from the digital compass was modified to filter out all data except the required heading data. The integrity of all of the data received was checked before it is passed on to the Radar WICO.

Radar WICO. The interfaces to the other two WICOs are defined above and attempt to test as many of the EM2P interface options as possible. The overall scheduling of the application was implemented so that the tracking algorithm runs after a full update of radar data has been received from the ACC radar. The request to the vehicle dynamics data was sent after the algorithm has run so that the next run of the algorithm has the latest vehicle dynamics and radar data. The GPS data was published from the GPS WICO totally asynchronous to the rest of the application.

The host vehicle’s GPS position was calculated in the Radar WICO using a Kalman filter algorithm that fuses GPS and Vehicle Dynamics data in order to update position at higher rate and overcome synchronism issues between the sensor WICOs. All of the targets reported from the ACC radar were in relative coordinates relative to the centre of the radar. These coordinates are then converted to the full GPS coordinate system and referenced to the tracked host vehicle position. This data was then available for fusing with other on board sensor data using the GPS co-ordinate system or for passing to the infrastructure for use in traffic management.

5.2.2. Experimental Results. To validate the WICOs in a “real-world” environment for this application it was decided that a test bench environment (Figure 13) would be used to playback a variety of recorded scenarios using data from the Radar, GPS, and Vehicle Dynamics devices to ensure consistency. Three runs of the same data were undertaken,

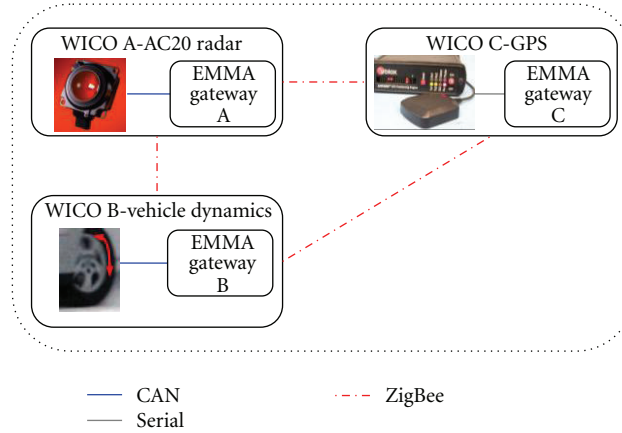


FIGURE 11: Vehicle level application diagram.

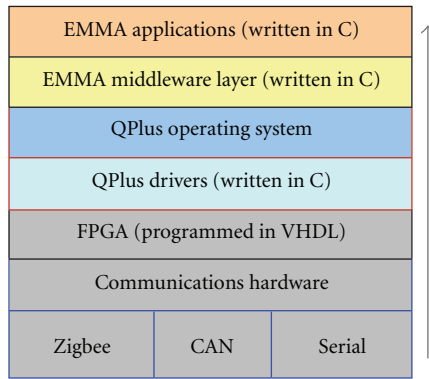


FIGURE 12: Application architecture on Xilinx board.



FIGURE 13: Three WICOs as tested in lab and in vehicle demonstration.

which would allow for the relevant metrics (message latency and lost messages) to be evaluated.

(a) Message Latency

Publish/Subscribe. It was only possible to check the timing of the publish/subscribe mechanism using a timestamp recorded in a log file. This was only set up to record to a one-second level of precision, and the GPS messages were only updated every second anyway. There was very little evidence

TABLE 7: Summary of lost message results for car-level communications.

WICO-WICO interaction	Missing messages		
	Run 1 (n, %)	Run 2 (n, %)	Run 3 (n, %)
GPS-Radar	15, 12.6	36, 28.1	16, 13.6
VehDyn-Radar	7, 1.2	9, 1.4	6, 1.0

of latency except in the third run where there was evidence some messages were delayed by one second.

Request/Response. The request/response time was measured internally in microseconds and reported in the log file. The results as shown in Table 6.

(b) *Lost WICO to WICO Messages.* The results showed in Table 7 a small message loss (around the 1% level for all runs) between the Vehicle Dynamics WICO and the Radar WICO, whereas there was a much higher message loss between the GPS WICO and the Radar, as high as 28.1% during Run 2.

There was minimal message loss between the Vehicle Dynamics WICO and the Radar WICO. It is supposed that these missed messages could have been caused by the Radar WICO simultaneously receiving a successful GPS message. However, comparison of log files proved inconclusive as all records of a missing Vehicle Dynamic message in the Radar logs coincided with a missing GPS message, which suggest a temporary total loss of communications between all WICOs. The only exception to this could be found in 4 records from all the Radar logs which had a missing Vehicle Dynamics message followed one second later by a successful GPS message, but these occurrences were not cyclical in the log files.

For the missing GPS messages, the causes for the higher loss rate were again not clear. A small number of messages in the GPS log files had a fault code which indicates that ZigBee communications were not allowed due to the system not acknowledging that the previous message had successfully

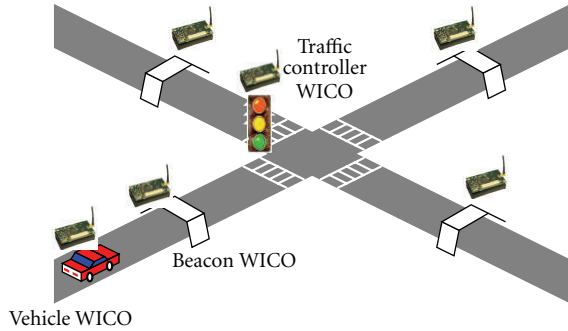


FIGURE 14: Giving priorities for emergency vehicle.

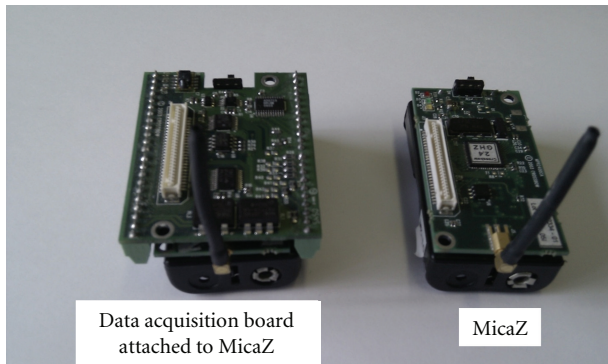


FIGURE 15: Micaz and MDA300.

been sent. Further investigation of these results is required to improve message lost between the Radar and GPS WICOs.

5.3. Supra-Vehicle Level Application. An application was developed in order to demonstrate the benefits of the middleware in priority to emergency vehicles. For that purpose, an emergency vehicle (ambulance, fire engine or police car, etc) would be equipped with a MicaZ WICO which would broadcast a beacon message if it was on an emergency mission (Figure 14). For example, in a busy intersection controlled by traffic lights, emergency vehicles are detected and given priority by regulating the state of the traffic lights.

5.3.1. Implementation. The implementation consists of two elements. The first element consists of a MicaZ WICO. The second element consists of a CITY traffic controller, manufactured by ETRA I+D, Spain. The CITY traffic controller is a well-proven controller that implements advanced capabilities for traffic management and control. A Cross-bow commercial data acquisition board MDA 300 (Figure 15) is used to provide as an interface between MicaZ WICO and CITY traffic controller with a small electric signal adaption stage.

In the demonstration, a MicaZ WICO was connected to a CITY traffic controller (Figure 16) that acted directly by providing information to the regulator about an emergency situation. Another MicaZ WICO was placed in the infrastructure to relay this message to the traffic light regulator.

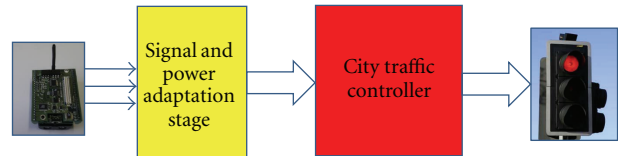


FIGURE 16: MicaZ to CITY Traffic Controller Interface.

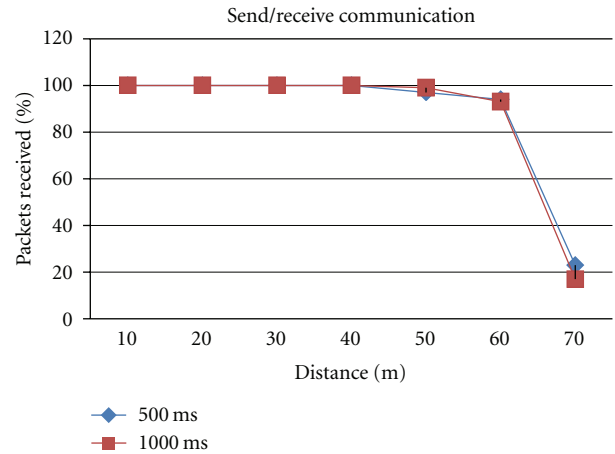


FIGURE 17: Urban environment packets received.

It was placed so far as needed in order to give time to the traffic regulator to change its status taking into account the time lost due to communication mechanisms (publish/subscribe) and other time periods the traffic regulator needs in order to guarantee safety first. The traffic regulator has been programmed to attend to the trigger signal provided by the mote activating an emergency control sequence. The demonstration was successfully carried out in a real road environment in Valencia, Spain.

5.3.2. Experimental Results. Several sets of experiments were carried out with EMMA middleware to evaluate possible use of the MicaZ WICO in the supra-vehicle level application. These experiments evaluated the use of MicaZ WICO with EMMA middleware for the application scenario at the supra-vehicle level. Two MicaZ WICOs (EMMA middleware running on them) were used for data-centric (request/response) and message-centric (send/receive) communication both in urban environment and mobile environment.

(a) Send/Receive Communication

Urban Environment Experiment. This experiment was carried out on Claremont Road, a busy road near Newcastle University. In each scenario, 100 packets were sent for every 500 ms, 1000 ms and those packets were received with another WICO which was connected to a Laptop via MIB 520 programming board. Both WICOs were placed at 1m above the ground. The MicaZ WICO's power level was set to default (NanoQplus power level 31). Each scenario was repeated three times, and calculations were performed offline to determine how many messages were lost at each distance and average values reported in Figure 17.

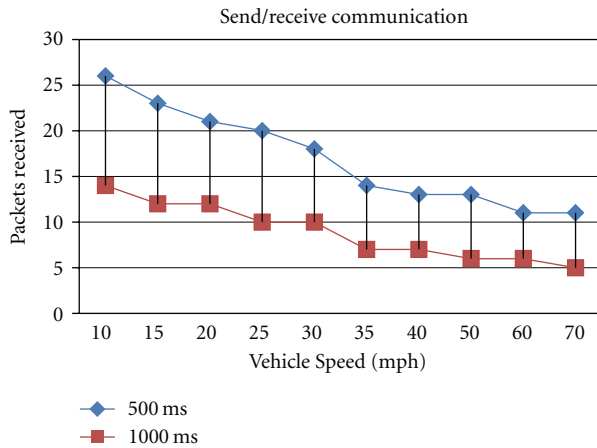


FIGURE 18: Mobile environment packets received.

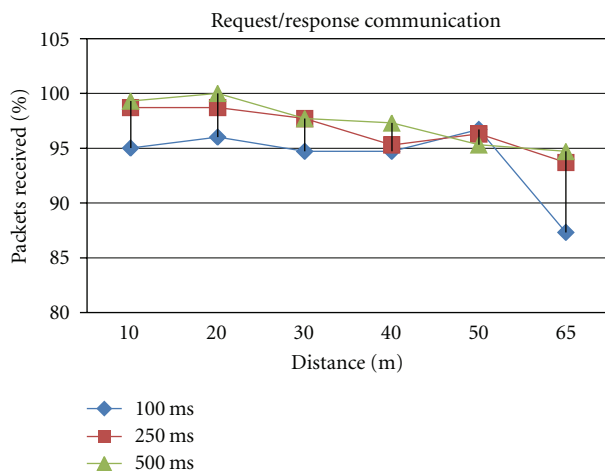


FIGURE 19: Urban environment packets received.

Mobile Environment Experiment. This experiment was carried out on Claremont Road up to 40 mph and on a Motorway near to the Newcastle Airport for higher speeds. The first MicaZ WICO was placed on a roadside stand 1 m from the ground, and the second MicaZ WICO was placed on the middle of the dashboard of a vehicle and connected to a Laptop via MIB 520 programming board. The MicaZ WICO at the road side sent messages periodically (500 ms, 1000 ms) which were received by the MicaZ WICO in the vehicle. Each scenario was repeated three times, and calculations were performed offline to determine how many messages were lost at each distance and average values reported in Figure 18.

In the mobile environment experiment, the received packets decreased with an increase in speed as the WICO is in range for a shorter period of time. This means that communication time window decreased with the increase in vehicle speed. At 70 mph speed, the WICO in the roadside received 5 and 11 packets for sending intervals 1000 ms, 500 ms, respectively. There were no packets lost between the first packet and the last packet received. This Experiment demonstrated that

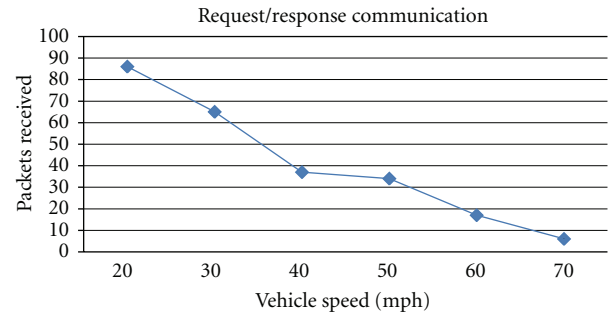


FIGURE 20: Mobile environment packets received.

MicaZ WICO can be used with EMMA middleware communication methods between a fixed infrastructure WICO and also fast moving vehicle-based WICO application. This is an important finding which proves that MicaZ WICOs do not suffer from any Doppler effects at normal motorway (70 mph) speeds.

(b) Request/Response Communication

Urban Environment Experiment. This experiment was carried out on Claremont Road, a busy road near to Newcastle University. Two WICOs were used for request/response communication with a request message transmitted every 100 ms, 250 ms, and 500 ms for different WICO-WICO separations from 10 m to 65 m. In each scenario, response packets were received and recorded. Both WICOs were placed at 1m above the ground. The MicaZ WICOs power level was set to default (NanoQplus power level 31). Each scenario was repeated three times, and calculations were performed offline to determine how many messages were lost at each distance and average values reported in Figure 19.

Mobile Environment Experiment. This experiment was carried out on Claremont Road up to 40 mph and on a Motorway near to Newcastle airport for higher speeds. The first MicaZ WICO was placed on a road side stand 1m from the ground and the second MicaZ WICO was placed on the middle of the dashboard of a vehicle and connected to a Laptop via MIB520 programming board. The MicaZ WICO at the roadside sent messages periodically which were received by the MicaZ WICO in the vehicle. Due to limited access to public roads and for safety reasons, the experiment was conducted only at a packet transmission interval of 100 ms. The experiment was repeated three times, and calculations were performed offline to determine how many messages were received at each distance and average values reported in Figure 20.

The urban environment experiment shows that packets can be received without any packet lost up to 45 m distance. The percentage of packets lost increases above 45 m distance in both cases. In the mobile environment experiment, the received packets decrease with the speed increases as the WICO is in range for a shorter period of time. This means that communication time window is decreasing with the

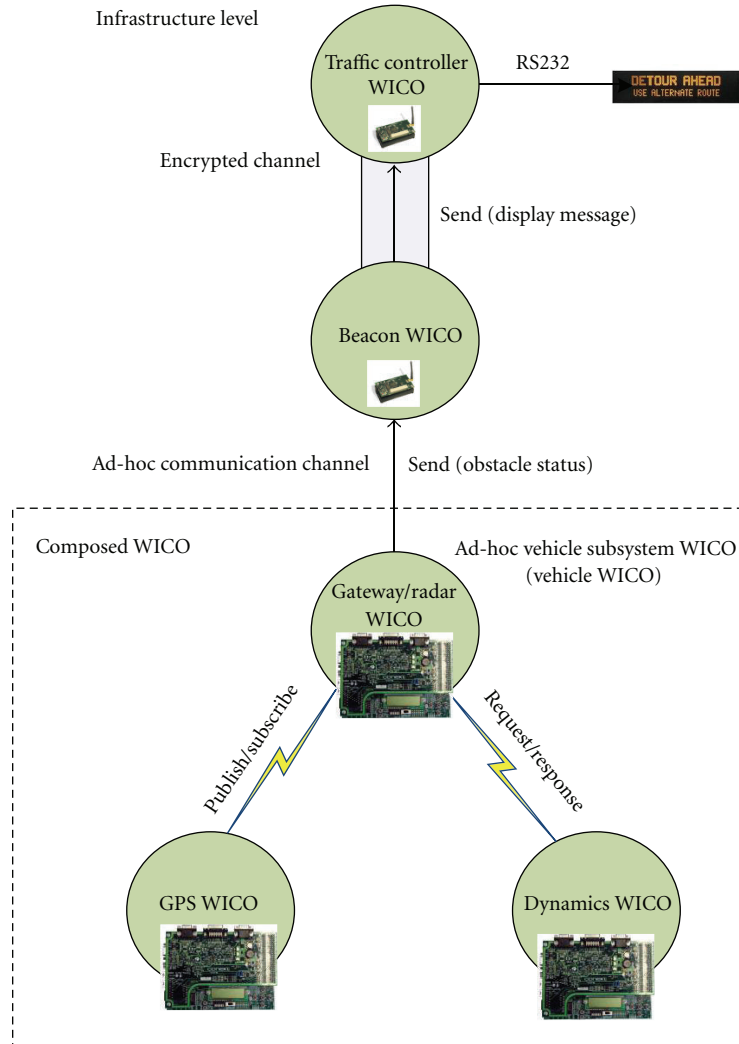


FIGURE 21: Deployment of WICOs in the inter-hierarchical demonstrator.

vehicle speed. At the 70 mph speed, The WICO in the roadside received 5 and 11 packets for sending intervals 1000 ms, 500 ms, respectively. And interestingly, there were no packets lost between the first packet and the last packet received. This experiment demonstrated that MicaZ WICO can be used with EMMA middleware communication methods between a fixed infrastructure WICO and also fast moving vehicle-based WICO applications. This is an important finding which proves that the MicaZ WICOs do not suffer from any Doppler effects at normal motorway (70 mph) speeds.

5.4. Inter-Hierarchical Level Application. One of the main objectives of the project was to achieve a middleware able to abstract complex subsystems formed by different kinds of WICOs into simpler elements (composed WICOs) that behave in the upper level system as a single unit. This way, complex applications could be built with a hierarchical shape, each group of WICOs working together on the same functionality appearing a unique element providing certain types of data to the remaining. In addition, the possibility to

form ad-hoc WICOs (i.e., to discover previously unknown elements on the system), and to propagate published data through the different abstraction layers, allowing their transformation and combination as it crosses certain points of the hierarchies, does really enhance the achievable possibilities of applications built on EMMA Middleware.

An inter-hierarchical application, integrating different hierarchical levels developed in the project: the car level and the supra-vehicle level, demonstrated how heterogeneity issues could be solved by developing middleware such as EMMA. In order to demonstrate the inter-hierarchical collaboration of the WICOs developed on the project, the application consisted of transforming the information provided by the vehicles at both automotive and vehicle subsystem levels into specific traffic control actions at infrastructure (i.e., supra-vehicle) level.

The inter-hierarchical demonstration made use of the WICOs at the vehicle level and supra-vehicle level. This demonstration aimed to provide advanced warning to a vehicle behind that there is an obstacle ahead. As can be seen

in Figure 21 the inter-hierarchical demonstrator made use of all communications mechanisms in the EM2P middleware and exercised most of the functionality of the middleware. In this demonstrator, inter-hierarchical collaboration of the WICOs developed on the project consisted of transforming the information provided by the vehicles at vehicle level (GPS, Vehicle Dynamics and Radar sensors based on Xilinx ML 403 platform with TRW Conekt daughter board) into specific traffic control actions at infrastructure (to MicaZ) level. In the infrastructure level, two MicaZ WICOs were used. First MicaZ WICO was used as beacon WICO to relay any message received by ad-hoc vehicle subsystem WICO to the second MicaZ WICO which was connected to a portable VMS panel to display the information sent by the vehicle level WICO. This application was successfully demonstrated for the EMMA project final review in a real road environment in London.

6. Conclusions

It is clear that the next generation of vehicles will be required to have increased safety, lower emissions, and more entertainment with higher performance than those of today. The innovations in wireless sensor devices will enable novel automotive applications which will become very common in future transportation applications. The challenges such as integrating heterogeneous wireless devices for specific transportation application can be met by developing middleware technologies such as in EMMA. This paper has presented the EMMA project that has been undertaken to investigate the suitability of using heterogeneous wireless sensors in transportation system applications. The validation of the prototype applications shows that wireless sensor networking technologies can be used at the engine level, vehicle level, and supra-vehicle level. The ability to communicate between vehicle and roadside illustrates that wireless sensor networks will enable efficient and discrete communications between vehicle and roadside.

Acknowledgments

Authors would like to thank Alberto Zambrano Galbis (ETRA I+D) for his contribution to the supra-vehicle level and inter-hierarchical level application. The work presented here was sponsored by EC under FP6 programme (IST-2005-5-034097) and authors also would like to thank all the project partners for their contributions.

References

- [1] The Eddington Transport Study, UK Department of Transport, May 2011, <http://webarchive.nationalarchives.gov.uk/+http://www.dft.gov.uk/about/strategy/transportstrategy/eddingtonstudy/>.
- [2] A. Tully, "Pervasive tagging, sensors and data collection: a science and technology review for the foresight project on intelligent infrastructure systems," *IEE Proceedings Intelligent Transport Systems*, vol. 153, no. 2, pp. 129–146, 2006.
- [3] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: a survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.
- [4] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, 2005.
- [5] X. Li, W. Shu, M. Li, H. Y. Huang, P. E. Luo, and M. Y. Wu, "Performance evaluation of vehicle-based mobile sensor networks for traffic monitoring," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1647–1653, 2009.
- [6] H. Cai and Y. Lin, "A roadside ITS data bus prototype for intelligent highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 344–348, 2008.
- [7] K. Selvarajah, A. Tully, and P. T. Blythe, "Integrating Smartdust into the Embedded Middleware in Mobility Applications (EMMA) Project," in *4th IET International Conference on Intelligent Environments, IE08*, Seattle USA, 21–22 July 2008.
- [8] T. Kosch, I. Kulp, M. Bechler, M. Strassberger, B. Weyl, and R. Lasowski, "Communication architecture for cooperative systems in Europe," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 116–125, 2009.
- [9] CAR 2 CAR Communication Consortium, July 2010, <http://www.car-to-car.org/>.
- [10] Zigbee Alliance, July 2010, <http://zigbee.org/>.
- [11] G. Cena, A. Valenzano, and S. Vitturi, "Advances in automotive digital communications," *Computer Standards and Interfaces*, vol. 27, no. 6, pp. 665–678, 2005.
- [12] K. Geihs, "Middleware challenges ahead," *Computer*, vol. 34, no. 6, pp. 24–31, 2001.
- [13] S. Hadim and N. Mohamed, "Middleware: middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, pp. 1–23, 2006.
- [14] TinyOS, July 2010, <http://www.tinyos.net>.
- [15] NanoQplus and Qplus, July 2010, <http://www.qplus.or.kr>.
- [16] Crossbow Technology: "MPR/MIB User's Manual", May 2011, <http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html>.
- [17] Xilinx, December 2010, <http://www.xilinx.com/>.
- [18] M. Peller and N. Berwanger, "Road vehicle Controller area Network (CAN)—part 4: time-triggered communication," ISO IS 11898.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

