A Sensor Based Peer to Peer Vehicle Data Sharing System, An Internet of Vehicles Approach

Reza Malekian[1,2], Arnav Thakur[1], Lakshmi Nair[1], Christian Fischer Pedersen[3]

[1] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa
[2] Department of Computer Science and Media Technology/Internet of Things and People Research Center, Malmō University, Malmō, 20506, Sweden
[3] Department of Engineering, Aarhus University, Aarhus, Denmark
reza.malekian@ieee.org, u11100037@tuks.co.za, lrnair2426@gmail.com, cfp@eng.au.dk

Abstract

This paper describes the work that has been done in the design and development of a vehicular peer-to-peer data sharing system, with the objectives of increasing the situational awareness of the motorist and to reduce or eliminate accidents. Sensors are used to detect objects along the road that a vehicle is travelling on. This information is then displayed to the motorist and warning messages are relayed to peer vehicles through vehicle to vehicle communication. To improve the situational awareness of the motorist, each vehicle can receive and send information to a road side unit, through vehicle to infrastructure communication. A central server remotely manages and monitors the overall peer-to-peer system. Qualification tests are conducted to validate various aspects of the system. The results indicate that the system is capable of vehicle to vehicle communication and vehicle to infrastructure communication communication for sharing information to prevent accidents and promote safe driving.

Keywords: Peer-to-peer, Vehicle-to-vehicle, Vehicle-to-infrastructure, Internet of Vehicle

1 Introduction

Vehicular accidents are a major global issue caused mainly by avoidable human error and improper driving practices. These include: not maintaining safe driving distances, misjudging the distance between vehicles, misinterpretation of road signs, disregarding blind spots and negligent driving [1-2]. The repercussions of such accidents include human injury or even death, damage to property, financial losses, etc. Also, in remote\rural areas, telecommunication is often limited and unreliable, making it difficult to report and get assistance from emergency services for locating hospitals and other public services infrastructures in the case of an accident or vehicular breakdown. It is thus essential to develop efficient and reliable vehicular accident prevention and handling mechanisms.

Existing technology involves vehicles mounted with distance sensors which assist in maintaining safe driving distances, lane keeping and identifying blind spots. However, this mechanism does not give warnings for vehicle side collisions, nor can it warn the driver of potential mistakes and improper driving of peer motorists who can also cause accidents.

In recent years, improvements to in-vehicle computing and processing capabilities, advancements in mobile and wireless communication and the wide spread implementation of the wireless sensor network (WSN) [3], has resulted in the emergence of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication.

V2V is a large-scale distributed system for wireless communication and peer-to-peer (P2P) information exchange between vehicles based on agreed communication protocols and standards [4]. V2V communication enables vehicles to maintain a safe driving distance, and send warnings to peer vehicles of potential problems, such as sudden braking, tyres bursting, etc. V2I is a bi-directional system where vehicles can request and receive traffic information and probe data to/from infrastructure [5]. V2I communication provides location based alerts, accident event related alerts, speed limit related information, precautions to be taken in poor weather conditions, road diversions, traffic disruptions etc.

With the advent of intelligent transport systems (ITSs), a popular approach to realizing V2V and V2I communications, is the vehicular ad-hoc network (VANET) [6]. In this de-centralized ad-hoc system, vehicles communicate with each other in a multihop way, and a vehicle broadcasts traffic conditions to other vehicles in its transmission range. It is still vulnerable to limited network capacity, scalability issues, and limited connectivity. Although VANETs incorporate state-of-art technology and communication

*Corresponding Author: Reza Malekian; E-mail: reza.malekian@ieee.org
DOI: 10.3966/160792642018121907018

standards, it will take significant time until the necessary market penetration of VANET technology is achieved.

In [7], a D4V system is developed for use with a smart phone to disseminate information V2I applications. The system uses a P2P overlay which unifies the concepts of geographical and virtual neighbourhoods.

A P2P cooperative caching scheme is introduced in [8]. Specifically targeted for high mobility, but sparse vehicular distribution environments, P2P data sharing with a hidden Markov model is implemented.

In [9], a network coding technique that cancels the interference caused by relay signals to vehicles that are receiving messages information centers, is proposed. The information spread of a joint V2I and V2V systems is considered.

Roadside units (RSUs) are considered expensive; thus, their deployment is very limited. In [10], a method of distributing the number of RSUs in a low-cost manner, yet efficient manner to better aide V2I communication.

In [11-12], propose systems combing V2V and V2I combination. In [11], a ZigBee and 802.15.4 based V2V and V2I system is presented.

In this paper, the design and development of a hybrid V2V and V2I system is presented. A control unit is designed for each vehicle containing distance sensors and communication interfaces. The distance sensors can detect objects. Communication occurs with a peer vehicle, and a special communication protocol is developed for this. And also with a road side unit (RSU), which provides location related traffic and weather information to vehicles and receives reports from these vehicles pertaining to alerts and warnings.

The following sections in this paper include Section 2 Methods, which contains the design and implementation of the solution developed in the project. Section 3 contains the results obtained from the design and implementation of the system. The paper is concluded in Section 4.

2 Methods

2.1 System Overview

The P2P system proposed in this study can be divided into three subsystems, as shown in Figure 1.

The first subsystem is an embedded system located in the vehicle and consists of the sensors and communication components. This vehicular subsystem (VS) can receive warnings from peer vehicles and issue warnings to peer vehicles. The VS can also receive location based alerts from the RSU.

The second subsystem is an embedded system located in the RSU. The roadside subsystem (RS), acts as the coordinator of the WSN managing the VS. The RS receives warnings and requests from motorists and transmits them to the CS which manages and processes the requests.

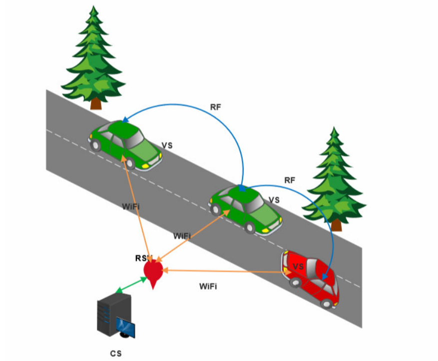

Figure 1. The subsystems of the P2P data sharing system

The third subsystem comprises the CS which manages and monitors the P2P vehicle data sharing system. The CS contains a graphical user interface (GUI) application and a web portal for managing the system remotely. The CS creates and issues alerts and precautions, which are transmitted first to the RS, then the VS.

The system is designed for operation in vehicles travelling up to a maximum speed of 120 km/h and a maximum speed limit of 15 km/h between peer vehicles. The V2V communication range is 10 m and the V2I communication range is 50 m.

2.2 The Vehicular Subsystem (VS)

2.2.1 The Hardware Design

The hardware components used to realize the subsystem are shown in Figure 2.

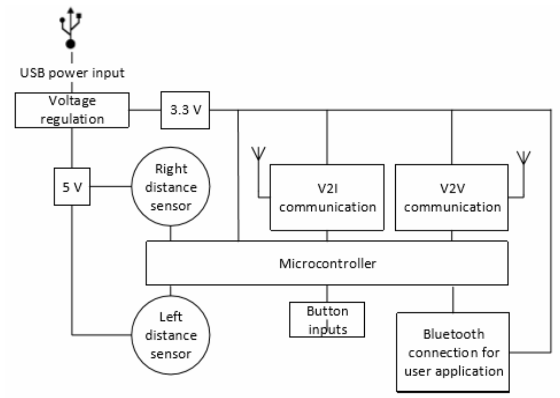

Figure 2. Block diagram of the components of the VS

The microcontroller used is the dsPIC30F4013 to handle input, process them and generate output.

A sound-based distance measuring technique, using ultrasonic waves, is implemented for measuring the distance of obstacles. It involves the measurement of time between the release of a sound wave and receiving the echoed wave, reflected by an obstacle. [13] This time difference, along with the speed of sound in the medium (air), is used to determine the distance to the obstacle. There is a linear relationship between the distance to the obstacle and the time taken to measure the distance. Ultrasonic sound wave based obstacle detection does not rely on the color, texture, or reflectivity of the obstacle, thus intensive calibration is not required to measure obstacles of various forms and in varied lighting conditions. An HC-SR04 non-contact ranging module consisting of an ultrasonic transmitter and receiver is selected as the distance measurement sensor for its high accuracy, small measurement angle of 15° and low cost, as shown in Figure 3.

Figure 3. The ultrasonic transmitter and receiver used for measuring distance

In the VS, three forms of communication occur, with peer vehicles (V2V), with the RS (V2I) and with the user application.

Communication with the user application involves short communication ranges of less than 5 m. Bluetooth (IEEE 802.15) is chosen as it operates in the 2.4 GHz frequency band, has moderate data rates of up to 25 Mbps, operates at a maximum range of 10 m and has low power consumption. The HC-06 Bluetooth module was used.

The (V2V) communication module is based on how fast connections can be established, low power consumption and wide accepted frequency range. The nRF24L01+ transceivers from Nordic semiconductors are identified as the most appropriate choice, as shown in Figure 4. The transceivers operate on the 2.4 GHz ISM band. The frequency can be chosen from 126 RF channels with a frequency range of 1 MHz making it ideal to be used when the source and receiver are in motion [14-15]. This is an advantage as interference can be avoided by configuring the module to operate on other programmable frequencies. Gaussian frequency shift keying (GFSK) based modulation scheme is used by the transceiver. Air data ranges from 250 Kbps to 2 Mbps can also be chosen. The output transmission power is also programmable. Low current usage for transmission of just 11.3 mA and 13.5 mA required for reception.

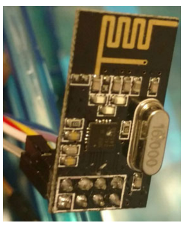

Figure 4. nRF24L01+ module used for V2V communication

The V2I communication is established using Wi-Fi. It is more suitable for V2I communication due to its longer range, support for multiple network topologies, higher data transfer rates, large number of supported connections, security and ability to establish TCP connections [16-19]. The ESP 8266 Wi-Fi module from Espressif systems is used.

2.2.2 Sensor Distance Calculation

As stated above, the distance measurement to an obstacle is given as follows:

$$Distance = \frac{speed \times time}{2}, \qquad \textbf{(1)}$$

where *speed* is the velocity of sound in the medium of air equivalent to 340 m/s and *time* is the time taken for the emitted wave to return.

The time taken for the emitted sound wave to return is measured using the timer module of the microcontroller. The timer is initialised to measure distances up to a maximum distance of 5 m. The pre-scaler value for the timer was chosen accordingly for measuring time intervals up to a maximum time of 30 ms required to measure distances of obstacles at a maximum range of 5 m. The pre-scaler value determines the factor by which the clock is divided. The microcontroller performs a single instruction for every four clock cycles, thus the instruction cycle frequency is 30 MIPS. The instruction cycle, *Tcy* is thus calculated as follows:

$$Tcy = \frac{1}{30 \times 10^6} = 33.33ns \qquad \textbf{(2)}$$

The time counted by a single counter timer tick is given as:

$$timer\ tick\_timelapse = Prescaler\ value \times Tcy \qquad \textbf{(3)}$$

The maximum time value that the timer can count for a given pre-scaler value before the counter overflows is:

$$Maximum\ time = Prescaler\ value \times Tcy \times 65535 \qquad \textbf{(4)}$$

where 65535 is the maximum 16-bit timer counter value before it overflows.

The timer interval is divided into more time steps for smaller pre-scaler values to improve accuracy. From the available pre-scaler values of 1:1, 1:8, 1:64 and 1:256, the smallest pre-scaler value for measuring a maximum of 30 ms is 1:64. The number of timer counts between the time of departure of the transmitted ultrasonic wave and the received wave determines the time taken to measure and is given by the following equation:

$$Measured\_time = Prescaler\ value \times Tcy \times TMR \qquad \textbf{(5)}$$

where *TMR* is the number of timer counts and the pre-scaler value is 64.

The timer 1 interrupt is configured using the associated period register, *PR1*. The *PR1* value can be calculated by the interrupt interval, pre-scaler value and the instruction cycle. A pre-scaler value of 64 is chosen with an interrupt interval of 5 ms, thus the distance to obstacles is measured once every 5 ms. The measuring frequency is 200 Hz which means that 200 distance readings are measured every 1 s. This interrupt interval is chosen as the maximum sideward displacement possible for an obstacle moving sideward with a velocity of 10 m/s (36 km/hr). The following equation calculates the *PR1* value:

$$\begin{aligned} PR1 &= \frac{Interrupe\ interval}{Prescaler \times Tcy} \\ &= (0.005)/(64 \times 33.33 \times 10^{-8}) \\ &= 2344 \end{aligned} \quad \textbf{(6)}$$

2.2.3 Communication Protocol

A communication protocol is developed for exchanging and interpreting message blocks sent between the subsystems. The protocol has the following message structure, as shown in Figure 5: the category of the message, destination subsystem, type of the message, location of the sender of the message and the content of the message. Each message block is separated by a "%" delimiter. The location marker is used by the motorist user application to verify the validity and relevance of the received packet. The message content is organized as a string containing the information identifier followed by the value which is processed by the receiver handlers.

```
Destination_marker % Location % message_type % Message_block %
```

Figure 5. The general message structure

2.2.4 Communication Interfaces

V2I communication is achieved with WiFi and communication between the VS and user application occurs with the use of Bluetooth. These interfaces are connected to UART1 and UART2 serial connection to the microcontroller.

The word strings received on the UART1 interface are processed using the communication protocol in the receive handler. The destination markers are decoded to identify the interface in which the packet needs to be forwarded. If the destination is *V*, the contents of the packet are saved to the data structures of the firmware and used when the external interrupts are invoked by the push buttons of the VS. If the destination marker is *P*, it is transmitted to the peer VS using the V2V communication channel. If the destination marker is *S*, it is transmitted to the CS using the V2I connectivity on UART2 with CIPSENDD AT command of the Wi-Fi communication module. The warnings detected by the VS through the distance measurement, is transmitted to the user application using the *U* marker from the UART 1 interface. An algorithm describing the above implementation is shown below.

Algorithm 1. receiver and transmitter communication handler related to the UART 1 interface

```
procedure comm_handler(message_str)
 extract destination_marker
 extract message_type
 extract message_block
   if destination_marker == 'V'
       message_block destined for embedded system
       firmware data structures populated
  else if destination_marker == 'P'
       Message is for peer motorist
       call V2V_communication procedure
        (message_block)
       return
  else if destination_marker == 'S'
       Message is for server
       call V2I_transmitter(message_block)
       return
   else if warning detected
       send message to user application using UART1
       call V2V_transmitter (location, message_block)
       return
   else
       return
end procedure
```

V2V communication is achieved with the nRF24L01+ RF wireless communication module, which interfaces with the serial peripheral interface (SPI) of the microcontroller.

The registers of the peer nRF modules need to be configured. A basic write operation is done by driving the chip select (CSN) pin low. The write command and hex address of the register associated with the configuration is written to the SPI interface. The write command in binary representation is "001A AAAA" where "A AAAA" is the 5-bit address of the register. The hex representation of the write command and the register address is written to the SPI interface. This is followed by writing the value to be set on to the SPI interface. The CSN pin is digitally set to 1. Basic transmit and receive functions are shown in the in Figure 6 and Figure 7 below.

```
void receive_NRF()
{
    PORTDbits.RD2=1;        // CE PIn
    PORTDbits.RD3 =0;      //CSN
    spi_write(0x61);      //RX_payload command

    int i=0;
    char tmp='1';
    while (tmp != '\r')
    {
        MSG_NRF[i] = spi_write(0);
        tmp=MSG_NRF[i];
        i = i + 1;
    }
    PORTDbits.RD3 =1;
    MSG_NRF[i]='\0';
    //place end of string character
}
```

Figure 6. Code snippet showing the receiving process

```
void transmit_NRF(char response[])  //send
{
  PORTDbits.RD3 =0;
  spi_write(0xA0);//TX_PAYLOAD
  int i;
  for ( i=0; i<strlen(response); i++)
    spi_write(response[i]);
  //Send carriage return
  spi_write('\r');
  spi_write('\n');
  PORTDbits.RD3 = 1;

  PORTDbits.RD2 = 1;
  for(i=0;i<350;i++)
      Nop();
  //delay_time(0.000010);
  PORTDbits.RD2 = 0;
  Nop();
}
```

Figure 7. Code snippet showing the transmission process

2.2.5 The User Application

The user application is the graphical display for interaction with the motorist in the VS. The application is designed in the Android environment and interfaces to the VS via Bluetooth. The interface displays warnings and location based alerts relevant to the user, including alerts from the peer motorist, server and the RSU. Side distance to obstacles and related warnings are also displayed. The speed limit and current speed is also displayed and the user is alerted when speed limits are exceeded. The user application also has buttons for sending a custom alert to the server and the peer VS.

The user application detects the speed of the vehicle and the sudden braking events using the accelerometer sensor of the user device. The detection of a sudden braking event generates a warning message for the peer and is transmitted to the VS through the Bluetooth interface. Accelerometer readings are taken every 10 ms and so the sampling frequency is 100 Hz. The x axis acceleration describing movement along the linear plane is used. The following equation of motion is used to calculate the speed of the vehicle.

$$v = u + \alpha\delta t \tag{7}$$

where v is the current calculated speed, u the previous speed reading, a is the acceleration and δt is the time difference of 10 ms between two readings.

The previous speed value is used for calculating of the current speed. If a speed difference of 10 m/s (36 km/h) is found in the 1 s time interval, a sudden brake warning is detected. The warning message for the peer is prepared using the communication protocol. The warning message packet is sent to the VS using the Bluetooth transmitter interface and is forwarded to the peer by the V2V communication channel.

2.3 The Roadside Subsystem (RS)

2.3.1 The Hardware Design

The microcontroller implemented in this subsystem is the dsPIC30F4013 and the WiFi communication with the VS is achieved with the ESP 8266 Wi-Fi module. The RS is connected to a console, running a road-side application, with a USB-to-Serial bridge.

The USB-to-Serial bridge is implemented using the MCP2200 USB 2.0 to UART converter. The 5 V power of the USB connector is connected to the voltage (VDD) of the integrated circuit (IC) chip. Figure 8 shows the connections used in the USB-to-Serial interface, where RXD and TXD are the serial outputs.

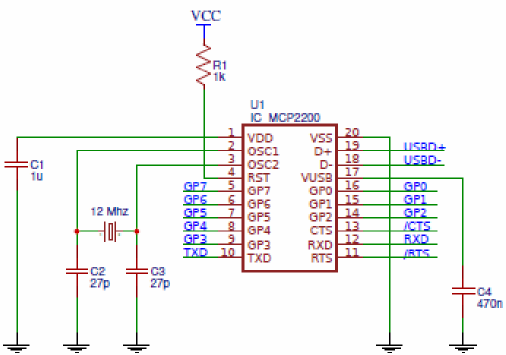

Figure 8. The USB-to-Serial interface circuit [20]

The USB-to-Serial Bridge operates at 5 V and the WiFi communication module operates at 3.3 V. The transmit line of the bridge is connected to the receive line of the communication module and vice versa. The transmit line of the USB (Tx_USB) operates at 5 V and the connected receive line of the communication module (Rx_comm) operates at 3.3 V. Thus, a logic level shifter is required and is done using a voltage divider. The following equation is used to calculate the values of the voltage divider.

$$\frac{Vout}{Vin} = \frac{Rx\_comm}{Tx\_USB} = \frac{3.3}{5} = \frac{R2}{R1+R2}, \quad \textbf{(8)}$$

The above equation is used to select values of the resistors *R1* and *R2,* as shown in Figure 9 below. The value of *R1* is chosen as 1 kΩ and *R2* is chosen as 2 kΩ.

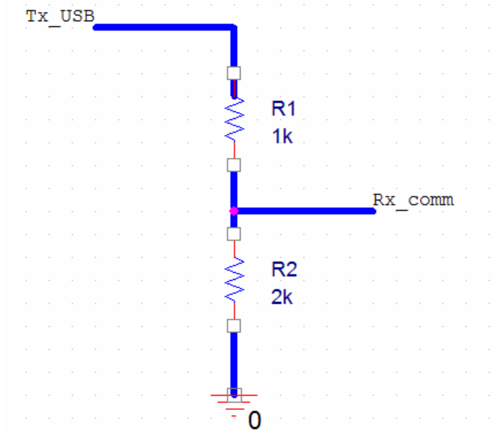

Figure 9. The logic level converter circuit

2.3.2 The Console Application

The console application created with. Net is used to monitor the data incoming to the RS as shown in Figure 10. SQL statements are used to update the list of RSUs [15-17] available at the start of the application.

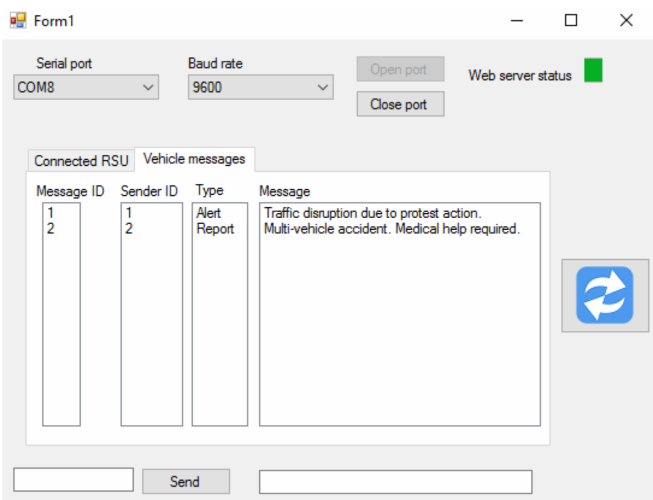

Figure 10. The console application of the RS receiving updates from vehicles

3 Results and Discussion

3.1 Range and Accuracy of the Distance Sensor

Experiments were conducted to determine the range and accuracy [13, 18-19] of the distance sensor. The distance measuring circuit was connected to the microcontroller and is used to measure the distance of an obstacle placed at a known distance. The obstacle used for this experiment was a metallic bar. The distance measuring code was executed in the debug mode of the IDE and timer counts are observed. The timer counts were used to compute the distance and the error is calculated.

The Table 1 summarizes the results obtained from the experiment.

Table 1. Summary of results obtained from the distance measuring experiment

Distance to obstacle (m)	Timer count(s)	Time for reflection (s)	Measured distance (m)	Error (%)
0.05	138	0.000294	0.050043	-0.08599
0.07	193	0.000412	0.069988	0.017618
0.1	276	0.000589	0.100086	-0.08599
0.12	331	0.000706	0.120031	-0.02555
0.5	1379	0.002942	0.500067	-0.01346
1	2758	0.005883	1.000135	-0.01346
2	5515	0.011764	1.999907	0.004667
3.3	9160	0.019539	3.321694	-0.65741
3.5	9651	0.020587	3.499746	0.007257
5	14064	0.03	5.100034	-0.10068

The system was designed to measure the distances of obstacles at a maximum range of 5 m. From the experiment it can be observed that the distances measured by the sensor are very close to the theoretical value and the differences are less than 10 %. Distance measured of an obstacle placed at an actual distance of 5 cm is underestimated by 0.086%. This increases to an error of 0.1% for obstacles at a distance of 5 m. Thus it can be observed that the distance sensor accurately measures the more sensitive smaller distances and further highlights the reliability of the distance sensor.

For an obstacle placed at a fixed location, the distance measuring function is called multiple times and on each call an output pin of the microcontroller is toggled and the output wave is examined on an oscilloscope. This is done to observe the consistency of the measurement functionality for the same distance. The uniformity of the output wave observed on the oscilloscope for an obstacle placed at a distance of 1 m is shown in Figure 11.

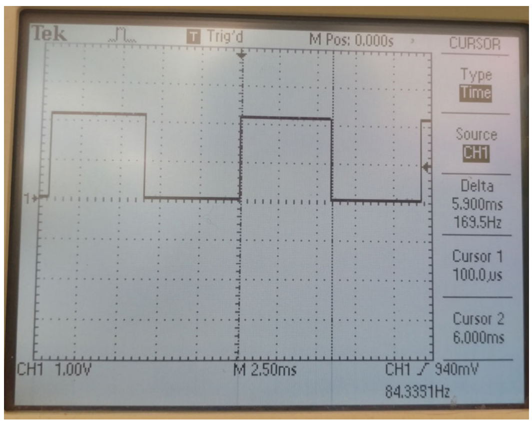

Figure 11. Toggled output wave for an obstacle placed at 1 m from the ultrasonic receiver

3.2 Performance of the V2V Communication Channel

The receiver V2V communication circuit was configured to resend received packets. The transmitter was connected to a computer to view results.

The transmitter sent packets containing 10 characters (10 bytes) at 10 ms intervals. The receiver then echoed the received packet. The transmitter compared the sent packet with the echoed packet and the performance of the V2V system was determined from this. This process was repeated 5 times.

The experiment was conducted in both stationary vehicles and in vehicles moving at a speed of 5 km/h and separated by a maximum distance of 10 m. The experiment was repeated 5 times in both cases. It was observed that for the scenario with stationary vehicles, of the 100 packets sent in 5 iterations, 96 packets were successfully received and thus the communication channel achieves an accuracy of 96%. For the scenario with the moving vehicles and at a speed difference of 15 km/hr, 95 packets out of 100 transmitted packets were successfully received. Thus, the communication channel achieves an accuracy of 95% for this scenario. It can be observed from the results of the experiment that the V2V communication link functions reliably. The summary of the results is shown in Table 2.

Table 2. Summary of results obtained for the V2V communication channel

V2V tests	Stationary receiver and transmitter	Moving receiver and transmitter
Total number of 10 bytes packets transmitted.	20 packets transmitted every 10 ms for 5 iterations.	20 packets transmitted every 10 ms for 5 iterations.
Total number of 10 bytes packets received.	96	95
Success rate of channel.	96%	95%

3.3 Performance of the V2I Communication Channel

The receiver RSU was configured to receive and handle incoming TCP connections and requests. The transmitter was connected to a client computer and a wireless connection was established with the RSU server.

The client computer was then placed at a range of 50 m from the RSU and was used to test the connectivity to the Wi-Fi network of the RSU circuit.

The results obtained, when a trace route operation to the default gateway (IP address of the RSU) was performed, showed that the client was directly connected to the RSU waypoint and there was only one hop in the route. The round trip time (RTT) of a sent packet was not more than 1 ms indicating that the link had low latency and high speed.

From the extended ping results, it can be observed that for a stationary client at a radial distance of 50 m from the RSU router, the average round trip time was 19 ms and had a maximum peak time of 83 ms. The communication link had a data loss of 5% in this scenario. It can be observed that the communication link performed satisfactorily with a small latency and had an acceptable packet loss.

The extended ping based tests for a client moving away from the RSU up to a maximum distance of 50 m was also conducted. The average round trip time was 32 ms and had a maximum peak time of 117ms. The communication link had a data loss of 7% in such scenarios. It can be observed that the communication link performed satisfactorily with a small latency and acceptable packet loss. The results for the V2I communication channel is summarised in the Table 3.

Table 3. Summary of results obtained for the V2I communication channel

V2I tests	Stationary client	Moving client
Total number of packets sent.	24	34
Total number of packets received.	21	29
Success rate of channel.	95%	93%
Minimum round-trip time	8 ms	8 ms
Maximum round trip time	83 ms	117 ms
Average round trip time	19 ms	32 ms

5 Conclusion

A system that can successfully detect an object on the road and warn peer vehicles and send alerts to a RSU is presented in this study.

The system can detect objects on the road up to 5 m with a maximum error of 0.1%. The distance sensor employed can thus be considered reliable for providing accurate information.

A communication protocol was developed to enable vehicles to communicate with their peers and with RSUs. Both V2V an V2I communication is achieved in this study with accuracies of between 93% and 96%.

Various applications are also developed to provide motorists with actual information and to better determine that the system is working as intended.

Possible future work can include the vehicle subsystem detecting potholes, uneven roads, etc. Sensors can be implemented in the RS to sense weather conditions, determine if there are any hazardous pollutant.

Acknowledgements

The research project was supported by the National Research Foundation (NRF) South Africa (IFR 160118156967 and RDYR160404161474), T-Systems and Intervate (a T-Systems Company), South Africa.

References

[1] A. Djajadi, R. Putra, Inter-Cars Safety Communication System Based on Android Smartphone, *IEEE Conference on Open Systems (ICOS)*, Subang, Malaysia,2014, pp. 12-17.

[2] Zhenkai Zhu, Lixia Zhang, Ryuji Wakikawa, Supporting Mobility for Internet Cars, *IEEE Communications Magazine*, Vol. 49, No. 5, pp. 32-37, May, 2011.

[3] M. L. Sichitiu, M. Kihl, Inter-vehicle Communication Systems: A Survey, *IEEE Communications Surveys & Tutorials*, Vol. 10, No. 2, pp. 88-105, July, 2008.

[4] N. G. Ghatwai, V. K. Harpale, M. Kale, Vehicle to Vehicle Communication for Crash Avoidance System, 2016 *International Conference on Computing Communication Control and automation (ICCUBEA)*, Pune, India, 2016, pp. 1-3.

[5] W. Cho, S. I. Kim, H. K. Choi, H. S. Oh, D. Y. Kwak, Performance Evaluation of V2V/V2I Communications: The Effect of Midamble Insertion, *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, Aalborg, Denmark, 2009, pp. 793-797.

[6] D. Rawat, B. Bista, G. Yan, S. Olariu, Vehicle-to-Vehicle Connectivity and Communication Framework for Vehicular Ad-Hoc Networks, *Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, Birmingham, UK, 2014, pp. 44-49.

[7] M. Picone, M. Amoretti, G. Ferrari, F. Zanichelli, D4V: A Peer-to-Peer Architecture for Data Dissemination in Smartphone-Based Vehicular Applications, *PeerJ Computer Science*, Vol. 1, No. 3, pp. 1-31, August, 2015.

[8] N. Kumar, J. H. Lee, Peer-to-Peer Cooperative Caching for Data Dissemination in Urban Vehicular Communications, *IEEE Systems Journal*, Vol. 8, No. 4, pp. 1136-1144, December, 2014.

[9] Q. Wang, P. Fan, K. B. Letaief, On the Joint V2I and V2V Scheduling for Cooperative VANETs With Network Coding, *IEEE Transactions on Vehicular Technology*, Vol. 61, No. 1, pp. 62-73, January, 2012.

[10] J. Barrachina, Road Side Unit Deployment: A Density-Based Approach, *IEEE Intelligent Transportation Systems Magazine*, Vol. 5, No. 3, pp. 30-39, July, 2013.

[11] P. Keeratiwintakorn, E. Thepnorarat, A. Russameesawang, Ubiquitous Communication for V2V and V2I for Thailand Intelligent Transportation System, *National Telecommunications Council International Conference*, Bangkok, Thailand, 2009, pp. 1-6.

[12] Buddhika R. Maitipe, U. Ibrahim, M. I. Hayee, Eil Kwon, Vehicle-to-Infrastructure and Vehicle-to-Vehicle Information System in Work Zones Dedicated Short-Range Communications, *Transportation Research Record Journal of the Transportation Research Board 2324*, pp. 125-132, December, 2012.

[13] B. Letswamotse, R. Malekian, C.-Y. Chen, K. M. Modieginyane, Software Defined Wireless Sensor Networks and Efficient Congestion Control, *IET Networks*, Vol. 7, No. 6, pp. 460-464, November, 2018.

[14] B. Lathi, *Modern Digital and Analog Communications*, Oxford University Press, 2016.

[15] B. B. Letswamotse, R. Malekian, C.-Y. Chen, K. M. Modieginyane, Software Defined Wireless Sensor Networks: A Review on Efficient Resources, Applications and Technologies, *Journal of Internet Technology*, Vol. 19, No. 5, pp. 1303-1313, September, 2018.

[16] J. Prinsloo, Accurate Vehicle Location System Using RFID, An Internet of Things Approach, *Sensors MDPI*, Vol. 16, No. 6, pp. 1-24, June, 2016.

[17] N. Ye, Y. Zhang, R. Wang, Vehicle Trajectory Prediction Based on Hidden Markov Model, *KSII Transactions on Internet and Information Systems*, Vol. 10, No. 7, pp. 3150-3170, July, 2016.

[18] N. Ye, Z. Wang, Y. Zhang, R. Wang, A Method of Vehicle Route Prediction Based on Social Network Analysis, *Journal of Sensors*, Vol. 15, pp. 1-10, April, 2015.

[19] Y. Zhang, N. Ye, R. Wang, A Method for Traffic Congestion Clustering Judgment Based on Grey Relational Analysis, *ISPRS International Journal of Geo-Information, MDPI*, Vol. 5, No. 5, pp. 1-15, article ID: 71, 2016.

[20] Microchip Technology Inc., USB 2.0 to UART Protocol Converter with GPIO, *MCP2200 Data Sheet*, pp. 1-32, March, 2010.

Biographies

Reza Malekian is with the Department of Computer Science and Media Technology/Internet of Things and People Research Center, Malmö University, Sweden and an Extraordinary Professor with the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria. His research is focused on Internet of Things, Sensors and smart cities. He is registered as a chartered engineer with the Engineering Council of the UK.

Arnav Thakur received the B.Eng. degree in computer engineering from the University of Pretoria, Pretoria, South Africa, in 2015, where he is currently pursuing the postgraduate studies with the Department of Electrical Electronic and Computer Engineering. His research interests include intelligent transport systems and advanced sensor networks.

Lakshmi Nair received her B.Eng. in computer engineering in 2004 and a M.Eng. degree in electronic engineering in 2009. She is currently pursuing her Ph.D. degree in computer engineering at the University of Pretoria, South Africa. Her areas of interest include wireless sensor networks, intelligent transport systems.

Christian Fischer Pedersen is an Associate Professor at Dept. of Engineering, Aarhus University, Denmark. His research interests include signal processing, machine learning, data analysis, and pervasive computing.