Vehicular Communication: Protocol Design, Testbed Implementation and Performance Analysis

Sumaiya Iqbal\textsuperscript{1}, Shihabur Rahman Chowdhury\textsuperscript{1}, Chowdhury Sayeed Hyder\textsuperscript{1}, Athanasios V. Vasilakos\textsuperscript{2} and Cheng-Xiang Wang\textsuperscript{3}
\textsuperscript{1} Dept. of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Bangladesh
\textsuperscript{2} Dept. of Computer and Telecommunication Engineering
University of Western Macedonia, Greece
\textsuperscript{3} School of Engineering and Physical Science
Heriot-Watt University, UK
sumaiya_iqbal@yahoo.com, shihab.buet@gmail.com, sayeed@csebuet.org
vasilako@ath.forthnet.gr, cheng-xiang.wang@hw.ac.uk

Abstract
Vehicular Communication Networks and Systems (VCNS) and Intelligent Transportation Systems (ITS) are one of the most attractive and challenging topics in recent days since a well efficient protocol for vehicular communication can facilitate the reduction of traffic congestion and can provide us with many more promising applications. In this paper, we propose a protocol for vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication. As one of the challenging parts of this paper, we present an experimental testbed in which two major applications of V2I & V2V communication (i.e. traffic congestion detection and emergency warning) is implemented. Based on careful analysis, we also calculate some key system parameters which reflect the efficiency of the protocol in different applications.

Categories and Subject Descriptors
C.2.1 [Computer Communication Networks]: Network Architecture and Design—Wireless communication; C.2.2 [Computer Communication Networks]: Network Protocols

General Terms
Design, Experimentation, Performance.

Keywords
Vehicular communication, message collision, data recovery, testbed, congestion, emergency warning.

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

\textsuperscript{410} IWCMC’09, June 21-24, 2009, Leipzig, Germany
Copyright © 2009 ACM 978-1-60558-569-7/09/06 ...$5.00.

VCNS is an emerging type of wireless network in which vehicles and roadside units are the communicating nodes; providing each other with information, such as safety warnings and traffic information. VCNS is different from other networks in the sense that the nodes are moving here. Mobility of the nodes creates various problems along with the problems of wireless network. Despite all these limitations, VCNS provides various services, for example, eliminating traffic collisions, ensuring smooth flow of vehicles in the highways, reducing traffic congestion, enabling a plethora of new applications such as mobile infotainment etc. This makes VCNS one of the most challenging topics for research and development in recent days. We are moved by these facts and try to put some contributions in this field.

The paper is organized as follows. In section 2, we discuss some previous works in this field. Section 3 gives an architectural overview of our system. Overview of our proposed protocol is discussed in section 4 and the implementation details are described in section 5. In Section 6, we present the experimental results. We also shed some lights on future works in section 7.

2. PREVIOUS WORKS

Since V2V and V2I communication has numerous applications in our real life, so it has been already studied extensively. Some researchers have proposed different protocols for communication. A channel access protocol for inter-vehicle communication was introduced in [9]. In this protocol channel is allocated to the mobile/stationary nodes based on their instantaneous geographical location. Authors in [10] introduced a new information propagation scheme for vehicular networks which makes use of attribute based data from MANET methodologies. Agarwal \textit{et al.} contributed in upstream message propagation in vehicular Ad Hoc networks [1].

Various kinds of applications have also been shown in previous days. A multi-hop wireless parking meter network (PMNET) to quickly find available parking space was proposed by researchers in [6]. Efficient protocol for secured vehicular communication and its impact on transport safety has been studied in recent days [7, 8]. Vehicle manufacturers are also showing their interest in VCNS. In order to improve driving safety, traffic organization and easy hotspot connec-
tions, six European car manufacturers founded the CAR 2 CAR Communication Consortium. Its goal is to create a European industrial standard for future communicating cars spanning all brands.

All these previous works regarding V2I and V2V communication have mainly focused on either theoretical aspects of communication protocol or on a particular application (mainly car parking). We do not limit our work just in developing a protocol; rather we design a testbed where we make real implementation of two important application scenarios (congestion detection and emergency warning). We also evaluate the performance of our implementation.

3. SYSTEM ARCHITECTURE

Our developed system consists of two modules:
1. Road Side Station Module (RSSM)
2. Vehicle Module (VM)

Road Side Station Module: RSSM is a static infrastructure and acts as a local server for a specific area. It communicates with the vehicles within its radio range. It has two parts: i) Broadcaster (BRD) ii) Responder (RSP)

Broadcaster is responsible for broadcasting necessary information to all the vehicles in the range of that RSSM. Responder is responsible for interacting with the vehicles (sends and receives necessary information) and also with the BRD to request it to broadcast any necessary information which is provided to it by RSP. It also stores information received from the vehicles for future decision making.

Vehicle Module: VM is embedded within the vehicles and so acts as moving nodes. It is responsible for communicating with other vehicles and RSSM and displays various information received from them.

Each part of every module in our system has its own Transceiver Unit (TU) and Control Unit (CU). TU is responsible for wireless communication and CU is responsible for controlling the overall operation. So each module of the system is independent of each other and performs their specific job. The block diagram of our system architecture is shown in Figure 1.

![Block Diagram of System](image)

Figure 1: Block Diagram of System

4. COMMUNICATION PROTOCOL

In this section, we propose Dual Channel Vehicular Communication (DCVC) Protocol for V2V and V2I communication. All communication is performed in two separate channels (ch1 & ch2). One channel is assigned only for broadcasting and another channel for other message transmission. This reduces message collision and increases reliability of communication.

In our protocol, we devise an effective method for detecting data loss as well as for data recovery, which uses the Binary Exponential Back off (BEB) Algorithm [2]. Redundant data transmission is also avoided whenever it is possible.

4.1 Framing

Framing of data is necessary for synchronization, error control, flow control etc. There are various methods of framing. Of them, Starting and ending flags, with bit stuffing is used here. The frame structure is shown in Figure 2.

![Frame Structure](image)

Figure 2: Frame Structure

The Starting Flag field contains a special bit pattern which is used to indicate the start of a new frame. The Control field is used for containing any pattern/number needed for synchronization. The Error Checking Code is used for error detection. The Type field identifies different types of frame. The Data field contains any information that is needed to be communicated.

The Ending Flag field contains the same special bit pattern to indicate the end of a frame.

4.2 Communication Rules

Each RSSM and VM is assigned a unique number as their ID (sID and vID) respectively. The BRD broadcasts its RSSM’s sID at regular intervals in ch 1 and the RSP always listens at ch 2. When a vehicle enters into the range of a RSSM, the VM receives the sID of that station. In response, VM sends the vehicle’s vID in ch 2 which is received by that RSSM’s RSP. Then for ensuring the proper communication between the RSSM and VM, RSP sends back an acknowledgement (ACK) to that vehicle in ch 2, which contains that vehicle’s vID.

Now, there are some special aspects of our protocol for improving its performance.

Since the RSSM always broadcasts its sID at regular intervals, it is possible that a vehicle within the range of a RSSM can get the same sID for multiple times. For this, the VM always saves the received sID. When it receives a sID, it compares it with the saved sID. If no match is found, the VM sends its vID and replaces the saved sID with the new one. Otherwise it does not transmit anything which reduces redundant data transmission.

Now, if multiple vehicles send their vID at the same time, at most one vID will be received by the RSP. Then, it sends an ACK containing the vID of the vehicle whose vID is received by it. So, only one vehicle gets proper ACK and other vehicles receive ACK which does not contain their vID. This is how other vehicles detect that a message collision has occurred and their data has been lost. Then for data recovery, they retransmit their vID after waiting a random time (in between 0 to 2^−1 ms for ith attempt) according to the BEB Algorithm [2]. The vehicles keep retransmitting until a correct ACK is received or the upper limit of the BEB Algorithm is reached.

To keep track of the vehicles within its range, the RSP always maintains a list of vehicles as well as the total count of
vehicles(Vehicle Count) that are currently within its range. If the RSSM is down for a while then the information is needed to be updated after it is powered up. And this information is also needed to be refreshed regularly(necessary to track the leaving of a vehicle from the range of a RSSM). For these purposes, the BRD broadcasts a special request when it is turned on and at regular intervals. When a VM receives this request, it must send its vID to the RSSM in any condition.

When a vehicle faces any emergency(e.g. accident, road blockage), it sends an emergency notification(EN) to RSSM in ch 2. Then, the RSP sends back an ACK to that vehicle using the same channel. The vehicle also broadcasts EN in ch 1 for a fixed ENC(Emergency Notification Counter) number of times. ENC is set to a value assumed to be enough to maximize the number of successfully notified vehicles and also to prevent the delay from being very high. The EN contains the vID of the vehicle in emergency and its current location(vID of the RSSM). The vehicles receiving the EN then forward it using ch 1 to their neighboring vehicles. To prevent EN being forwarded infinitely, it contains a value as TTL(Time To Live) which is decremented in each hop and when it becomes zero that EN is discarded. To prevent duplicate EN to be forwarded during this process, each VM maintains a list of vIDs received by an EN. This list is reset in regular short intervals. When a VM receives an EN it looks up the vID within it in the list. If a match is found then the message is considered to be a duplicate one which is then discarded.

For these purposes, the BRD broadcasts a special request when it is turned on and at regular intervals. When a VM receives this request, it must send its vID to the RSSM in any condition.

When a vehicle faces any emergency(e.g. accident, road blockage), it sends an emergency notification(EN) to RSSM in ch 2. Then, the RSP sends back an ACK to that vehicle using the same channel. The vehicle also broadcasts EN in ch 1 for a fixed ENC(Emergency Notification Counter) number of times. ENC is set to a value assumed to be enough to maximize the number of successfully notified vehicles and also to prevent the delay from being very high. The EN contains the vID of the vehicle in emergency and its current location(vID of the RSSM). The vehicles receiving the EN then forward it using ch 1 to their neighboring vehicles. To prevent EN being forwarded infinitely, it contains a value as TTL(Time To Live) which is decremented in each hop and when it becomes zero that EN is discarded. To prevent duplicate EN to be forwarded during this process, each VM maintains a list of vIDs received by an EN. This list is reset in regular short intervals. When a VM receives an EN it looks up the vID within it in the list. If a match is found then the message is considered to be a duplicate one which is then discarded.

5. EXPERIMENTAL TESTBED

Our testbed has one RSSM and three VMs. Without loss of generality, we assume that this model of one RSSM and three VMs simulates a real life scenario.

5.1 Hardware Implementation

The hardware components used in our system are: a) AVR ATmega8 microcontroller b) Flash writer c) RFM12 FSK Transceiver d) 7805 Linear voltage regulator e) Antenna etc. Details of these components can be found in [3, 5, 4].

We have used radio frequency as physical medium for wireless communication.

The key component of the CU is an ATmega8 microcontroller. Two interfaces of the microcontroller are used: SPI( Serial Peripheral Interface ) and USART( Universal Synchronous Asynchronous Receiver Transmitter ) [3]. The flash writer is used to write programs to ATmega8’s ROM.

The key component of TU is RFM12 FSK (Frequency Shift Keying) radio transceiver. For transmitting data the internal TX register and for receiving data built in FIFO register of the RFM12 module is used. The end of transmission is checked by the RGIT bit of the status register and the completion of data reception is checked by the FFIT pin of RFM12 module. A 17.6 cm jumper wire is used as RFM12 module’s antenna. A range of up to 100 m(radius) is achieved by this antenna [5]. A 7805 linear voltage regulator is used to prevent any voltage glitch that can harm the performance of RFM12 module.

We have used 434MHz and 439.75MHz frequencies of the 433MHz free ISM band as ch 1 & ch 2 respectively. All the data transmission has been performed in 4800bps.

5.1.1 Road Side Station Module

BRD is equipped with a radio transceiver( RFM12 module ) and a microcontroller( ATmega8 ) and they are connected through SPI of ATmega8.

RSP is equipped with similar components as the BRD. In addition, a 7-segment display is also connected with the microcontroller for showing various information.

The internal communication between BRD and RSP is carried out using strobe i/o and USART.

The block diagram of RSSM is shown in Figures 4.

5.1.2 Vehicle Module

VM is equipped with a radio transceiver( RFM12 module ), a microcontroller( ATmega8 ) which are connected through SPI. Here, the microcontroller is also connected with two 7 segment displays for showing various information and a switch to simulate different emergency events( e.g. accident, road blockage ).

The block diagram of the VM can be found in Figure 5. The complete snapshot of RSSM and VM in our testbed is shown in Figure 6.

5.2 Implementation Details

Here, we describe some implementation specific issues which are used in our experimental testbed.

The format of the frame in our implementation follows the basic frame structure proposed in the DCVC protocol described in section 4. The frame format is given in Figure 7.
Starting Flag and Ending Flag: A 24bit special bit pattern "0xAAAAAA" is used as the starting and ending flag. This is a requirement for the RFM12 module [5].

Error Checking Code: This field contains an even parity bit which is calculated based on the bits of Type and Data field for error checking.

Type: There are 5 types of frames in our implementation which is identified by the Type field of the frame.

- **Type 1[000]: Station Broadcast(SB) frame**: This frame is used to broadcast sID of RSSM. RSSM broadcasts it once in every 0.1s.

- **Type 2[001]: Vehicle Response(VR) frame**: This frame is used by VM to send its vID to RSSM.

- **Type 3[010]: Acknowledgement(ACK) frame**: RSSM sends acknowledgement to VM using this frame.

- **Type 4[011]: Emergency Notification(EN) frame**: VM sends EN message to RSSM and other vehicles within its range using this frame.

- **Type 5[100]: Station On(SO) frame**: This frame is broadcasted by RSSM when it powers up and also after every 5s to refresh its stored information.

Data: Data field is divided into two subfields: i)Vehicle Info ii)Station Info. Each contains different information according to frame type which is listed in Table 1.

### Table 1: Contents of Data field

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Vehicle Info</th>
<th>Station Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>SB</td>
<td>Vehicle Count</td>
<td>sID</td>
</tr>
<tr>
<td>VR</td>
<td>vID</td>
<td>sID</td>
</tr>
<tr>
<td>ACK</td>
<td>vID</td>
<td>sID</td>
</tr>
<tr>
<td>EN</td>
<td>vID</td>
<td>sID</td>
</tr>
<tr>
<td>SO</td>
<td>Vehicle Count</td>
<td>sID</td>
</tr>
</tbody>
</table>

6. EXPERIMENTAL RESULTS

In this section, we perform some tests to evaluate the efficiency of our protocol. Here, we consider two popular applications - congestion detection and emergency notification. During these tests we measure some time intervals as key parameters. For this, the modules are interfaced with a computer using parallel port and necessary software is written in Java.

6.1 Application 1: Congestion Detection

We have used the number of vehicles to approximate the congestion status of an area. Whenever a vehicle enters into the range of a RSSM, it is immediately informed about the Vehicle Count of that area. So the driver can take decision whether to go through that area or avoid it.

6.1.1 Implementation Issues

When a vehicle enters into the range of a RSSM, the RSSM and VM communicates according to the DCVC protocol described in section 4.2. The vehicle list is maintained by the RSP. The BRD gets the Vehicle Count from the RSP through serial communication over USART interface of AT-mega8. This Vehicle Count is broadcasted with each SB and SO frame in every 0.1s and 5s respectively. After broadcasting the SO frame, the BRD notifies RSP to reset the vehicle list through strobe i/o. So a vehicle entering the range of a RSSM gets the the current Vehicle Count in that area which is shown in the display attached with the vehicle.

6.1.2 Performance Analysis

To evaluate the protocol performance, we define some system parameters and perform some tests under different conditions to measure them for our experimental setup.

<table>
<thead>
<tr>
<th>Condition</th>
<th>No. of VMs</th>
<th>RT_{rssm}(avg)(ms)</th>
<th>Increase of RT_{rssm}(avg)%</th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>1</td>
<td>15</td>
<td>168.8667</td>
</tr>
<tr>
<td>b)</td>
<td>2</td>
<td>19</td>
<td>189.2941</td>
</tr>
<tr>
<td>c)</td>
<td>3</td>
<td>20</td>
<td>208.75</td>
</tr>
</tbody>
</table>

Table 2: Result Summary for RT_{rssm}
indicates the time required for other vehicles to track a new vehicle into the range of the RSSM.

<table>
<thead>
<tr>
<th>Condition</th>
<th>No. of Test Cases</th>
<th>RT\textsubscript{vm(avg)} (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) 3 vehicle in system</td>
<td>20</td>
<td>219.6</td>
</tr>
</tbody>
</table>

Table 3: Result Summary for RT\textsubscript{vm}

Table 2 and 3 summarize the result for RT\textsubscript{rssm} and RT\textsubscript{vm} respectively. Note that, with the increase of vehicles RT\textsubscript{rssm(avg)}'s rate of increase is diminishing. And also RT\textsubscript{vm(avg)} for same condition(3 vehicles in system) is greater than that of RT\textsubscript{rssm(avg)}. This is due to the rule of our protocol that information about a new vehicle is received first by RSSM and then it is broadcasted to the VM\textsubscript{s}. We also plot the RT\textsubscript{rssm} for different number of vehicles in our system in Figure 8. Since, we have three VM\textsubscript{s} in our testbed we use Logarithmic Interpolation to determine the trend of data for 100 VM\textsubscript{s}.

6.2 Application 2: Emergency Warning

In our system, whenever a vehicle encounters an accident it immediately sends this information to the RSSM and also broadcasts this message to all other vehicles in its range.

6.2.1 Implementation Issues

Without loss of generality, we assume that whenever an emergency event occurs the detection mechanism of the vehicle generates a high voltage level to inform the VM about it. Here, we use a switch to simulate an emergency event.

When a vehicle faces an emergency, it takes necessary steps described in section 4.2. In our testbed we have set the value of ENC to 20. Due to hardware limitation, the vehicle in emergency broadcasts EN but further forwarding by the vehicles that receive that EN is not implemented. In our testbed this EN is shown in the displays by the letter “C” along with the vID of the vehicle that faced the accident and sID of the station to indicate the area where the accident occurred.

6.2.2 Performance Analysis

For this application, we also perform some analysis to calculate some system parameters that show the performance of our protocol.

RT\textsubscript{vm} (Response time for receiving EN at vehicles): It indicates the time required for the VM\textsubscript{s} to track an emergency event within its range(V2I communication).

<table>
<thead>
<tr>
<th>Condition</th>
<th>No. of Test Cases</th>
<th>RT\textsubscript{vm(avg)} (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) 1 vehicle in system</td>
<td>21</td>
<td>110.1129</td>
</tr>
<tr>
<td>b) 2 vehicles in system</td>
<td>20</td>
<td>388.7</td>
</tr>
<tr>
<td>c) 3 vehicles in system</td>
<td>15</td>
<td>634.2667</td>
</tr>
</tbody>
</table>

Table 4: Result Summary for RT\textsubscript{vm}

Table 4 and 5 summarize the result for RT\textsubscript{rssm} and RT\textsubscript{vm} respectively. However, the RT\textsubscript{rssm(avg)} and RT\textsubscript{vm(avg)} could also be plotted in a similar way as RT\textsubscript{rssm} in Figure 8 for different number of vehicles.

We implemented both the applications on our testbed. Some snapshots of our work is shown in Figure 9.

6.3 Frame Transmission Rate

Here, we find out Successful Frame Transmission Rate (SFTR) for V2I communication using the DCVC protocol. SFTR\textsubscript{V2I} (SFTR for V2I communication): This is a measure of how efficiently our protocol can successfully transmit data and handle data loss by message collision. SFTR is calculated by using the following formula:

\[
SFTR = \frac{nSF}{nTF} \times 100 \% \quad (1)
\]

\[
nSF = \text{Number of successfully transmitted frames}
\]

\[
nTF = \text{Total number of transmitted frames}
\]

The result for SFTR\textsubscript{V2I} is summarized in Table 6. We plot SFTR\textsubscript{V2I} for different number of vehicles in Figure 10. And it is seen from the plot that with increasing number of vehicles the SFTR\textsubscript{V2I} is reducing at a very slow rate which shows that our protocol is highly effective in successfully transmitting the frames.

6.4 Speed Limit for Vehicles

Here, we calculate the speed limit of the vehicles which allows them to stay within the range of RSSM so that the
Experimental testbed, we have used the free ISM band and low bitrate due to resource limitations. But 802.11 MAC sub layer implemented hardware could easily be used for high frequency and bitrate which would definitely improve the protocol’s performance.

7. FUTURE ENHANCEMENTS

There are lots of scopes to improve our protocol and enhance the application of it on which we are still working. For example, according to our protocol, each RSSM stores all information received from the vehicles within its range. A Central Server can be introduced which can communicate with multiple RSSMs and collect the information from them along with the timestamps of the events. Then this centrally stored information can be used for different purposes. Such as, tracking the location of a vehicle at any time(can be helpful to police force, private car owner), providing quick help to the vehicle in emergency by knowing its location etc.

8. CONCLUSION

In summary, we introduce an effective protocol for V2I and V2V communication and test it on a testbed. We also implement two of many possible applications of the protocol and characterize the performance of it.

9. REFERENCES