

Low cost approach to real-time vehicle to vehicle communication using parallel CPU and GPU processing

GOH CHIA CHIEH

Faculty of Electronic and Electrical Engineering
University of Nottingham
Selangor, Malaysia

DINO ISA

Faculty of Electronic and Electrical Engineering
University of Nottingham
Selangor, Malaysia

Abstract— This paper proposes a novel Vehicle to Vehicle (V2V) communication system for collision avoidance which merges four different wireless devices (GPS, Wi-Fi, ZigBee® and 3G) with a low power embedded Single Board Computer (SBC) in order to increase processing speed while maintaining a low cost. The three major technical challenges with such combinations are the limited system bandwidth, high memory requirement and slow response time during data processing when accessing various collision avoidance situations. Collision avoidance data processing includes processing data for vehicles on express ways, roads, tunnels, traffic jams and indoor V2V communication such as required in car parks. Effective methods are proposed to address these technical challenges through parallel Central Processing Unit (CPU) and Graphic Processing Unit (GPU) processing. With this, parallel V2V trilateration and parallel bandwidth optimization, multi-dimensional real time complex V2V data streaming can be attained in less than a second. The test results have shown that there is at least a 4 to 10 times improvement on processing speed with parallel CPU and GPU processing used in V2V communication depending on different road safety conditions.

Keywords-component; CUDA; Parallel processing; Vehicle to Vehicle Communication; WLAN; ZigBee.

I. INTRODUCTION

Road safety has drawn worldwide attention due to an increasing number of accidents every year. According to the United Nations (UN) road safety report approximately 90% of road fatalities occur in low and middle income countries [1] and high portion of these accidents are due to issues directly related to the driver.

Due to its high impact, much research on Vehicular Ad-Hoc Networks (VANET) and the safety applications that go along with it have been carried out to prevent or lower the rate of accidents [2] [3] [4] [5]. VANETs are the customary implementation of network communications for Intelligent Transport Systems (ITS). Dedicated Short-Range Communication (DSRC) is the standard used for V2V communication operating in 5.9 GHz. The most basic V2V communication system consists of GPS and General Packet Radio Service (GPRS) which transmit the current vehicle location to the nearby vehicle using mobile the Point to Point (P2P) network protocol.

VANET is designed to target for quicker response of the low latency network V2V communication connectivity. However as the infrastructure of the vehicles are getting more complex and the flow of traffic is increased, more sensors and electronic devices are built on-board the vehicle to enhance road safety conditions. Much research and the accompanying solutions have been implemented to lower down the cost of replacing the wireless V2V communication system which uses expensive transceivers with commonly used wireless network systems such as Wi-Fi (802.11), Wireless Broadband (802.16) ZigBee® (802.15.4) or other emerging network device [6] [7] [8] [9] [10]

The objectives of this research are as follows:

- **LOW COST** -To design a low cost V2V collision avoidance system through parallel CPU and GPU processing based on widely available GPS, Wi Fi, ZigBee® and 3G broadband units.
- **LOW POWER, HIGH SPEED** - Development of parallel bandwidth, data optimization and V2V trilateration algorithm, able to process the data in real time on a low power SBC.

II. RELATED WORK

Vehicular ad-hoc networks (VANET) have gain wide popularity since the mid-90s [11]. VANET is an important element for accident prevention for on road vehicle. Data retrieval and clear presentation to the driver from the surrounding environment has been proven to reduce human errors while driving [12] [13]. However, the wireless bandwidth for vehicle communication system is very limited [14] [15]. An efficient communication protocol is needed to avoid overloading the system.

In view of this, [16] [17] [18] [19] et. al proposed the GPS and GPRS/GSM communication which tracks the location of the vehicle. The data is uploaded to the server and is monitored in real time. The main weakness for the system is that, the data is transmitted or received through GPRS/GSM which only allows a small amount of data to be transferred in and out the system.

[20] [21] [22] [23] et. al proposed the V2V communication by implementing the Collision Warning Systems (CWS) which

are radar, camera and radio based. Radar/Ladar and camera-based collision detection is meant for detecting and viewing what is in front and at the rear of the vehicle. Their proposed method uses a radio based system which covers the side angle where a potential collision might occur. Radio based methods adopt a similar concept such as Wi-Fi stack - Time Division Multiple Access (TDMA). The main drawback for such setup is the system cost increases due to the implementation of Radar.

[24] [25] [26] [27] et. al proposed the Wireless Access for Vehicular Environment (WAVE) for V2V and Vehicle to Infrastructure (V2I) system. The proposed system uses the Road Side Unit (RSU) coordination scheme for WAVE safety services support. This means that, the Access Point (AP) or known as WAVE provider is set up along the road. The economic drawback of this method is the need to set up the high cost RSU's and this limits the number of locations that is covered.

Our work is focused on creating a cost effective real time V2V network system. This system in each vehicle consists of a Global Positioning System (GPS) satellite based positioning system and a wireless location positioning system (LPS) which was developed using low cost short range wireless ZigBee® device (IEEE 802.15.4). 3G broadband is used to transmit or receive the information from the server in case of emergencies such as vehicle breakdown, traffic jam or road accidents. The methodologies for such configurations are implemented through parallel CPU and GPU processing. This method is capable of not only boosting the performance occupancy but also reduces the delay latency on data transfer between hardware and software in an efficient order.

III. PROPOSED WORK

As shown in Fig. 1, is a top down embedded system architecture view. It consists of 5 layers, Red, Green, Blue, Purple and Orange. The system can be categorized into two parts: real time data processing and non-real time data processing. The real time requirement for data processing such as with the Wi Fi, ZigBee® and GPS will be given a higher priority as compare to Broadband transmission and speech recognition. Non real time data processing

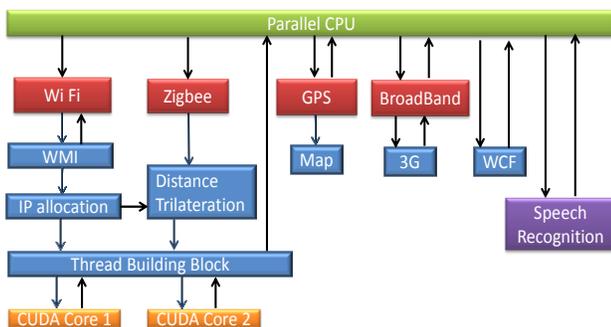


Figure 1. System Architecture

A. The Red and Green Layer

This is the hardware layer where Wi Fi, ZigBee®, GPS and Broadband Modem (represented in the block diagram) are connected and data is obtained from the real world. Data synchronization between devices was previously performed at the Green layer - Task parallelism library (TPL). TPL is one of the features in Microsoft® .net framework. The objective of using TPL is to divide the task into sections and parallel processed through the CPU. The incoming data from different devices were split into four different tasks. Each task was split into multiple threads. Threads can be parallel processed through multiple Central Processing Unit (CPU) and Graphic Processor Unit (GPU). If different data streams from the hardware devices simultaneously, data processing priority was given from the left to the right, which means the Wi Fi device has the highest priority and the Broadband device has the lowest.

Both Wi Fi device and ZigBee® consist of two parts, with a total of four devices. There are two devices in Wi Fi block diagram: Wi Fi router and Wi Fi client adapter. The Wi Fi client was installed on individual vehicle. Data transfer between Wi Fi client adapters of different vehicle is facilitated through the Wi Fi router. The maximum transmission range for the Wi Fi device is 100 meter radius wide, supporting up to 54Mbps of data transfer rate.

ZigBee® system consists of two items: the ZigBee® Coordinator and the ZigBee® router. The ZigBee® router is used to establish connection between vehicles in a mesh topology the total number depending on the ZigBee® system standard for number of devices within a mesh. Although the maximum data transfer rate for ZigBee® device is 250Kbps, it has a very low packet overhead during transfer. The data transfer rate will maintain its consistency within 100 meter radius. ZigBee® devices were used in V2V communication mainly for range estimation. ZigBee® devices are very low in power consumption. Hence, the device will therefore continue to operate even when the vehicle engine is turned off.

For outdoor location tracking, the Global Positioning System (GPS) is used. The GPS device follows the NMEA 0183 format for location and time zone retrieval. The fundamental problems for GPS are: the distance estimation tolerance is up to ± 15 meters [37] and GPS can only work in outdoor environment. Therefore, the combination of ZigBee® and GPS is a necessity to solve the problem.

The 3G broadband device is used for internet connection to send or download the latest update from the centralized management server. This information includes traffic info and accident location from the vehicle itself (if any).

B. The Blue Layer

The software layer, represented in Blue Layer, is the interface to the Hardware layer (Red Layer). Vehicle to vehicle communication was facilitated using Windows Communication Foundation (WCF) framework, which is a replacement for Winsock API.

Windows Communication Foundation served as the Transmission Control Protocol/Internet Protocol (TCP/IP) stack to support communication with the router. Each vehicle has its designated Internet Protocol (IP) address and a unique ID tag. The ID tag can be transmitted through ZigBee® device from vehicle to vehicle in a star topology that used multicast protocol. The IP address is generated in random order; it follows the IPV4 format, Class C network ranging from IP address 192.168.9.4 to 192.168.9.254. IP address which range from 192.168.9.1 to 192.168.9.3 is reserved for debugging and testing purposes. If IP conflict occurs, Windows Management Instrument (WMI) will assign a new IP to the current vehicle. The vehicle to vehicle communication is within 100 meters radius and has maximum of 30 vehicles linked within the range. Each time a communication is established, the router search through the network IP polls (192.168.9.4 to 192.168.9.254) for unique ID. Once the ID was identified, the properties of the vehicles, including the speed, location and the ID, are sent through the wireless communication. Once the vehicle is out of the 100 meter radius range, the IP will be released automatically so that the IP address will be available for another incoming nearby vehicle.

Without GPS, the location of the vehicle is calculated by using the distance trilateration method. This method pinpoints the vehicle location by comparing the distance of the 3 nearby vehicle. Once the location of the nearby vehicle location is identified, the vehicle location is computed through the Thread Building Block (TBB) [39]. TBB allows the thread to be parallel processed on two different GPU cores. Whenever the GPS signal is available, the map will be updated accordingly to reflect the latest location of any vehicle within the 100 meters vicinity radius range.

The vehicle ID, location, speed, time and date will be sent to the server through 3G broad. The server will help to monitor the traffic flow. If any accident happened in the particular location, the server will transmit the data to alert the driver.

C. The Purple Layer

Speech recognition was located at the purple layer. It was integrated into the system based on the built-in function from Windows OS. The system is programmed to detect extreme changes in voice pitch, e.g. shouting. The system has some pre-programmed command such as “send it”, “retrieve it” that allows user to interact with the system while driving. Once the system detects an abnormal tone, the system will switch to panic mode and take control of the dangerous situation automatically. The system will give audible warning such as “warning, nearby vehicle is within the range of xx meters”. Information regarding the vehicle, for example the location and the vehicle ID, will be transferred to the server as needed.

D. The Orange Layer

The orange layer represents the Graphic Processing Unit (GPU) parallelism layer. Compute Unified Device Architecture (CUDA) programming model has two parts: Host and Device. The Host is the CPU and the Device is the GPU. Data were transferred from the Host (CPU) to the Device (GPU) in a single-program, multiple data (SPMD) for preprocessing. Thread building block [39] transfers the array of data from the parallel CPU to the CUDA core 1 and CUDA core 2 for

parallel processing. The device will send the final results back to the Host. Standard C++ compiler is used for compiling data in the host, while CUDA’s own ANSI C based compiler is used for compiling in the device (GPU) coding.

IV. METHODOLOGY

A. Received Signal Strength Indicator (RSSI) Trilateration to find vehicle location.

Trilateration was used to locate a vehicle in the vicinity of three other vehicles. This was done in order to monitor the relative movement of vehicles surrounding each other to predict occurrences of abrupt evasive action such as when blind spots are encountered which leads to dangerous situations. This was accomplished by monitoring wireless signal strength. GPS was not used in this case because it cannot be used in indoor (car park) or in a sheltered operation.

The basic concept of wireless signal strength is shown Fig. 2. There are two wireless devices, as the two devices getting further apart, signal strength within the radius will be getting weaker. The measurement of the signal strength which received from the device is known as Received Signal Strength Indicator (RSSI). RSSI measures the power which received from the antenna in decibel meter (dBm). Thus, higher positive RSSI value indicators stronger signal strength.



Figure 2. Received Signal Strength

Based on the signal strength, there are 3 common methods to measure the distance between two points: Received Signal Strength Indicator (RSSI), Time of Arrival (TOA) and Angle Of Arrival (AOA). The simplest and cheapest method is the Received Signal Strength (RSS) method as it doesn’t required additional hardware.

As shown in Fig. 3, in order to estimate a location, at least three wireless devices were needed to triangulate the position of the 4th wireless device. There are three wireless devices, a, b and c. Wireless device d is within the intersection of a, b and c. The coordination of a, b and c can be obtained from either the Wi Fi or a fix point ZigBee® router from the vehicle.

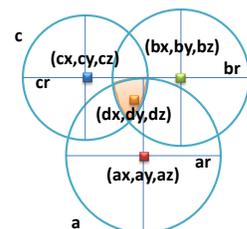


Figure 3. RSSI trilateration

As shown in Fig. 3 above, the equation of the trilateration can be written as:

$$\begin{bmatrix} (x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2 \\ (x_b - x_d)^2 + (y_b - y_d)^2 + (z_b - z_d)^2 \\ (x_c - x_d)^2 + (y_c - y_d)^2 + (z_c - z_d)^2 \end{bmatrix} = \begin{bmatrix} r_a^2 \\ r_b^2 \\ r_c^2 \end{bmatrix} \quad (1)$$

Where

r_a, r_b, r_c = radius of circle a, b and c

$x_a, x_b, x_c, y_a, y_b, y_c, z_a, z_b, z_c$ = x, y and z of GPS coordination of circle a, b, c and d

The intersection point of these three circles can be anywhere in the shaded orange area. In order to minimize the estimated error, the least square error method can be used:

$$\begin{bmatrix} [x_a - (x_d + Wd_x)]^2 + [(y_a - (y_d + Wd_y))]^2 + [z_a - (z_d + Wd_z)]^2 \\ [x_b - (x_d + Wd_x)]^2 + [(y_b - (y_d + Wd_y))]^2 + [z_b - (z_d + Wd_z)]^2 \\ [x_c - (x_d + Wd_x)]^2 + [(y_c - (y_d + Wd_y))]^2 + [z_c - (z_d + Wd_z)]^2 \end{bmatrix} \quad (2)$$

$$- \begin{bmatrix} r_a^2 \\ r_b^2 \\ r_c^2 \end{bmatrix} = \begin{bmatrix} E1^2 \\ E2^2 \\ E3^2 \end{bmatrix}$$

$$\text{Square Error} = E1^2 + E2^2 + E3^2 \quad (3)$$

Where

Wd_x, Wd_y, Wd_z = weight for dx, dy, dz coordination

From equation (2), the equation can be expanded as:

$$A = x_a^2 - 2[x_a(x_d + Wd_x)] + (x_d + Wd_x)^2 + y_a^2 - 2[y_a(y_d + Wd_y)] + (y_d + Wd_y)^2 + z_a^2 - 2[z_a(z_d + Wd_z)] + (z_d + Wd_z)^2 - r_a^2 \quad (4)$$

$$B = x_b^2 - 2[x_b(x_d + Wd_x)] + (x_d + Wd_x)^2 + y_b^2 - 2[y_b(y_d + Wd_y)] + (y_d + Wd_y)^2 + z_b^2 - 2[z_b(z_d + Wd_z)] + (z_d + Wd_z)^2 - r_b^2 \quad (5)$$

$$C = x_c^2 - 2[x_c(x_d + Wd_x)] + (x_d + Wd_x)^2 + y_c^2 - 2[y_c(y_d + Wd_y)] + (y_d + Wd_y)^2 + z_c^2 - 2[z_c(z_d + Wd_z)] + (z_d + Wd_z)^2 - r_c^2 \quad (6)$$

By using Newton-Raphson method to find the trilateration of circle a, b and c:

$$F = \begin{bmatrix} A(x, y, z) \\ B(x, y, z) \\ C(x, y, z) \end{bmatrix} \quad \Delta = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

$$F' = \begin{bmatrix} \frac{\partial A(x, y, z)}{\partial x} & \frac{\partial B(x, y, z)}{\partial y} & \frac{\partial C(x, y, z)}{\partial z} \\ \frac{\partial A(x, y, z)}{\partial x} & \frac{\partial B(x, y, z)}{\partial y} & \frac{\partial C(x, y, z)}{\partial z} \\ \frac{\partial A(x, y, z)}{\partial x} & \frac{\partial B(x, y, z)}{\partial y} & \frac{\partial C(x, y, z)}{\partial z} \end{bmatrix}$$

Where

$$F + F' \Delta = 0 \quad \Delta = -\frac{F}{F'}$$

$$\Delta x = -\frac{F(x)}{F'(x)} \quad \Delta y = -\frac{F(y)}{F'(y)} \quad \Delta z = -\frac{F(z)}{F'(z)}$$

$$x_{n+1} = x + \Delta x \quad y_{n+1} = y + \Delta y \quad z_{n+1} = z + \Delta z$$

The initial root value $x_{n+1}, y_{n+1}, z_{n+1}$ of function F is chosen arbitrarily at the beginning. Every n iteration, check the absolute value of gx, gy and gz:

$$gx = ABS \left[-\frac{F(x)}{F'(x)} \right] \quad gy = ABS \left[-\frac{F(y)}{F'(y)} \right] \quad gz = ABS \left[-\frac{F(z)}{F'(z)} \right]$$

The algorithm will stop at n iteration if gx, gy and gz value ≤ 0.1 . As gx, gy and gz get closer to 0, the estimated vehicle location gets closer to the coordinate of dx, dy and dz.

B. Bandwidth Optimization

1) Media Access Control (MAC) Address Lookup Table (LUT)

In order to implement the technique discussed previously, the ZigBee® was used to initiative data transfer from vehicle to vehicle and the WiFi network was used to transfer higher volumes of data. The Zigbee® has a lower overhead and therefore establishes the connection faster while the WiFi has a larger overhead (slower connection) but is able to handle higher amounts of data. Hence by using both in parallel we obtain a better performance.

Bandwidth optimization for the ZigBee® and WiFi systems is important because of the amount of data transfer which occurs from the surrounding vehicles at any given time. This amount adds uncertainty to the system hence optimization is needed. As this information is also processed in real time to predict dangerous situations, bandwidth optimization is a necessary part of feasibly implementing the trilateration technique discussed previously. In order to optimize the bandwidth usage for the Wi Fi, a Lookup Table (LUT) is used to map the MAC addresses of the Wi Fi router with ZigBee® coordinator and ZigBee® router.

As previously mentioned the WiFi network is used to transfer trilateration data between vehicles. Wi Fi was designed for data streaming and the maximum data streaming bandwidth from one Wi Fi device to another is up to 54 Mega bit per second (Mbps). However, establishing a connection between a Wi Fi client and a Wi Fi router connection requires some time.

As shown in Fig. 4, OpenWRT Wi Fi router is configured to be in a mesh topology. Wi Fi routers (yellow dot) can take up to 5 second to establish a single connection from the client (purple dots). It is more common to have the Wi Fi router to be fixed permanently on a particular spot, for example inside a building whereby the clients join the same Wi Fi router network.

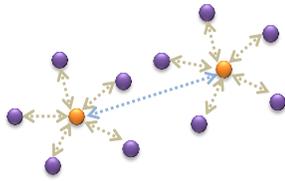


Figure 4. Wi Fi router and Wi Fi Clients

In this V2V communication system, there are 5 devices: GPS, ZigBee® coordinator, ZigBee® router, Wi Fi client and Wi Fi router. Each device of every vehicle has a unique MAC address and the same MAC address cannot be repeated.

Based on the Wi Fi 802.11 standards [40] Wi Fi MAC address can be obtained whenever the Wi Fi wireless signal is transmitted. The signal contains information on the Network Type, Authentication, Encryption and Basic Set Service Set Identification (BSSID).

BSSID has divided into 5 sections: wireless MAC address, Signal Strength, Signal Type and Data Transfer rate represented in Mega bit per second (Mbps).

The Wi Fi router wireless MAC address format is as follow:

CC: XX: XX: XX: XX: XX

CC = country code from 00- FF (Hexadecimal), representing 256 different countries based on a unique country code ID store on the server.

XX = Wi Fi MAC address (expressed in 2 bytes Hexadecimal from 00 – FF)

Since the Wi Fi router’s MAC address is unique for each vehicle, the following LUT is generated:

CC: XX: XX: XX: XX: XX, YY: YY: YY: YY: YY: YY: YY: YY, ZZ: ZZ: ZZ: ZZ: ZZ: ZZ: ZZ: ZZ

Where

YY= ZigBee® coordinator MAC address

ZZ= ZigBee® router (node) MAC address

2) *Data Transfer Optimization*

The wireless distance from one device to another is directly proportional to the rate of change of data transfer. In other words, the further the distance between vehicle 1 and vehicle 2 is, the weaker the wireless signal will be. Weak wireless signal limit less data to be transmitted or received from vehicle 1 to vehicle 2.

In order to find the optimal data transmission distance between vehicle 1 and vehicle 2, data optimization method is implemented to the system. As shown in Table I, the data transmission occurs only when the Wi Fi signal strength is greater than 10%. If the signal is low, the data which has the highest priority will be transmitted.

TABLE I. WI FI SIGNAL STRENGTH

Wi Fi Signal Strength (%)	Types of data to be transmitted
Less than 10%	X
10% to 29%	1
20% to 39%	2
40% to 59%	3
60% to 79%	4
80% to 100%	5

The data transfers are as follows:

1. Transmit the GPS coordination for maximum of 5 vehicle within 20 meters Wi Fi range
2. Scan for MAC address of ZigBee® coordinator and router of any vehicle nearby, once it found, linked up all the nodes together so that data can be transferred from one vehicle to another. Scanning stop when it reaches 10 vehicles or GPS coordination with Wi Fi radius of 40 meters (whichever comes first).
3. Scan and established the coordination between Wi Fi and ZigBee®, for any nearby vehicle. Stop scanning after it reaches 20 vehicles or Wi Fi range of 60 meters (whichever comes first)
4. Scan and established the connection of up to 25 vehicles up to 80 meters Wi Fi radius.
Scan and established the connection of up to 30 vehicles up to 100 meters Wi Fi radius

C. *Parallel CPU and GPU Implementation*

The Parallel Central Processing Unit (CPU) and Graphic Processing Unit (GPU) improve overall system performance with greater processing speed. It is crucial to implement the parallelization threads in a proper manner. In order to obtain an optimal result with parallelism, the system is relies strongly on the combination of CPU and GPU hardware performance strength, specifically favoring different type of memories for prioritize memories sequential access [41]. Currently, the limitation on CPU and GPU bandwidth become the bottleneck for hardware acceleration. There are limited bandwidth data to compute per cycle, CPU and GPU have to wait for the cycle to complete before proceeding. Compute Unified Device Architecture (CUDA), is a multithreaded GPU programming library designed by NVidia® to perform general purpose computation on a GPU. The GPU threads are executed through kernel [42]. A kernel has similar function to C programming function; the only difference is that the kernel function is requested from the CPU to the GPU device. The host (CPU) will sent the matrix array of data to the device (GPU) for arithmetic calculation. The final arithmetic value is then transferred back to the host.

1) *Parallel CPU and GPU Implementation*

The system is implemented in two folds. As shown in Fig. 1, there are 4 tasks handled by WLAN, ZigBee®, GPS and 3G broadband. Firstly prioritize the task. Among them, both WLAN and ZigBee® have the highest priority, followed by GPS and 3G broadband. Secondly, split the tasks into multiple threads. By using “divide and conquer” technique, each thread is partitioned into multiple parts. At the maximum performance occupancy, each part will not have more than 512 threads.

Lastly, the data matrix is sent to GPU for parallel calculation. Once the final answer has been obtained from the parallel GPU, it is transfer back to CPU for further processing.

Table II shows the 4 tasks implemented using Parallel CPU and GPU or Parallel CPU processing.

TABLE II. PARALLEL TASK DISTRIBUTION

Task	Parallel CPU + GPU	Parallel CPU
Task 1 (Bandwidth optimization)	✓	
Task 2 (GPS location transmission)		✓
Task 3 (3G broadband data transmission)		✓
Task 4 (vehicle location trilateration)	✓	

TABLE III. PSEUDO CODE (TASK 1 – BANDWIDTH OPTIMIZATION):

1.	Create a task and name it as <i>Task1</i> using Task parallel Library (one of the .net framework 4.0 features)
2.	Search nearby vehicle Wi Fi router for BSSI wireless MAC address, if different BSSI wireless MAC address is found, go to step 3, else go to step 8.
3.	Allocate the matrix array in GPU device
4.	Store the WiFi MAC and LUT to global kernel of the GPU
5.	Compare the WiFi MAC to match the LUT
6.	Once a match is found, transfer the match back to CPU host. If not, continue with Step 6. If no match is found, go to Step 2.
7.	Connect the wireless client based on the match, check for wireless data transfer optimization algorithm to optimize the data transfer bandwidth. If the WiFi router is further than 100 meters radius, release the IP for the next vehicle. If any IP address collision, reconfigure a new IP address.
8.	Go back to step 2.

TABLE IV. PSEUDO CODE (TASK 2 – GPS LOCATION TRANSMISSION):

1.	Create a task and name it as <i>Task2</i> using Task parallel Library (one of the .net framework 4.0 features)
2.	Check for CPU usage if it is more than 95%, if true, delay step 2 until progress it is less than 95%.
3.	Send the NMEA code to the GPS device to get the coordination of x, y and z. Get the amount of satellites, maximum 12. At the same time get the date and time.
4.	Broadcast the GPS coordination to the wireless network within the same transmission range.
5.	Check for incoming data from other vehicle, if the distance is too close, hidden vehicle at winding road; activate speech audio to warn drivers.

TABLE V. PSEUDO CODE (TASK 3 – 3G BROADBAND DATA TRANSMISSION):

1.	Create a task and name it as <i>Task3</i> using Task parallel Library (one of the .net framework 4.0 features)
2.	Check for CPU usage if it is more than 95%, if so, delay step 2 until progress it is less than 95%.
3.	Check whether the 3G wireless broadband is more than 80% and is stable. If not don't transmit to the server. (Step 3 is to avoid data transmission error)
4.	Check whether current vehicle stop at certain location for a certain time period at middle or side of the road. If so transmit the coordination to the server, the vehicle could be having an accident or break down on the road.

2) *Two dimension Newton-Raphson using parallel CPU and GPU*

Used in the trilateration calculation, the Newton-Raphson algorithm parallel CPU and GPU optimization steps are as follow:

$$\text{Radius of circle } r_a, r_b, r_c \text{ in meters} = 10^{(\text{RSSI}-A+X\sigma)/10n} \quad (7)$$

Where

A = RSSI value of distance in 1 meters = -40dBm

Xσ = Value of random noise = 0

n = path loss component value from 2 to 4.

From Equation (4), (5), (6):

Assume

$$Wd_x, Wd_y, Wd_z = 0$$

$$\text{Jacobian Matrix} = J = \begin{bmatrix} \frac{\partial A(x, y, z)}{\partial x_d} & \frac{\partial B(x, y, z)}{\partial y_d} & \frac{\partial C(x, y, z)}{\partial z_d} \\ \frac{\partial A(x, y, z)}{\partial x_d} & \frac{\partial B(x, y, z)}{\partial y_d} & \frac{\partial C(x, y, z)}{\partial z_d} \\ \frac{\partial A(x, y, z)}{\partial x_d} & \frac{\partial B(x, y, z)}{\partial y_d} & \frac{\partial C(x, y, z)}{\partial z_d} \end{bmatrix} = \begin{bmatrix} -2x_a + 2x_d & -2y_a + 2y_d & -2z_a + 2z_d \\ -2x_b + 2x_d & -2y_b + 2y_d & -2z_b + 2z_d \\ -2x_c + 2x_d & -2y_c + 2y_d & -2z_c + 2z_d \end{bmatrix} \quad (8)$$

$$\text{Inverse matrix of } J = J^{-1} = \frac{1}{\text{Determinant } J} * C^T$$

$$\text{Adjoin of } J = C^T = \begin{bmatrix} -2x_a + 2x_d & -2x_b + 2x_d & -2x_c + 2x_d \\ -2y_a + 2y_d & -2y_b + 2y_d & -2y_c + 2y_d \\ -2z_a + 2z_d & -2z_b + 2z_d & -2z_c + 2z_d \end{bmatrix}$$

$$\text{Determinant } J = -2x_a + 2x_d [(-2y_b + 2y_d)(-2z_c + 2z_d) - [(-2z_b + 2z_d)(-2y_c + 2y_d)]] - [(-2y_a + 2y_d)[(-2x_b + 2x_d)(-2z_c + 2z_d) - [(-2z_b + 2z_d)(-2x_c + 2x_d)]] + [-2z_a + 2z_d][(-2x_b + 2x_d)(-2y_c + 2y_d) - [(-2x_c + 2x_d)(-2y_b + 2y_d)]] \quad (9)$$

$$\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} - \left[(J^{-1}) \begin{pmatrix} A \\ B \\ C \end{pmatrix} \right] \quad (10)$$

*Note: The initial value for x_d, y_d, z_d were assigned randomly on the first iteration.

TABLE VI. PSEUDO CODE (TASK 4 – VEHICLE LOCATION TRILATERATION):

1.	Create a task through Task parallel Library (one of the .net framework 4.0 features) and name it as <i>Task4</i> .
2.	Within Task4 parallel function, check for the coordination of the GPS from nearby vehicle, if no GPS signal can be found, take the RSSI value only.
3.	Split the Task into two threads using Thread Building Block so that it can be parallel process using two GPU core at the same time.
4.	Allocate the array of 100 x 100 matrix array in GPU
5.	Allocate the block and threads in GPU global memory <ol style="list-style-type: none"> Compute the RSSI to distance value using equation (7)

b. Substitute the coordinate value of $x_a, y_a, z_a, x_b, y_b, z_b, x_c, y_c, z_c$ into equation (4), (5) and (6)

c. Compute Inverse matrix of J from equation (9)

d. Allocate the share memory 10x10x10 of GPU: `shared_memory[threadIdx.x][threadIdx.y][threadIdx.z]`

e. Initial guess value for $\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} = \begin{pmatrix} 200 \\ 200 \\ 200 \end{pmatrix}$ While $\begin{pmatrix} A \\ B \\ C \end{pmatrix} \geq \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$

i. Check weather is it the first time running the While loop, if so take the initial value of $\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix}$ Step e) or else skip step i and proceed straight to step ii.

ii. `shared_memory[threadIdx.x][threadIdx.y][threadIdx.z]` = Compute the equation (10) using synchronous threads $\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix}$

iii. move the value to

6. Transfer the final answer of x_n, y_n, z_n from the GPU back to CPU
Compute the final coordinate x_n, y_n, z_n of current vehicle location and transmit to other vehicle using Windows Communication Foundation (WCF).

V. RESULTS AND DISCUSSION

A. Software Setup

The following software in Table VII is used to compile and implement the proposed parallel system:

TABLE VII. SOFTWARE SETUP

Software	Methodology/Algorithms used in this research
CUDA 4.1 [28]	Parallel GPU processing for WiFi and ZigBee® transmission protocol and bandwidth optimization.
Visual Studio 2010 (C# and C++) [29]	C# for handling non time critical task, that is graphical user interface (GUI). C++ for handling time critical real time task WiFi and ZigBee®, GPS and 3G broadband interface
Windows Communication Foundation 4.0 [30]	V2V WiFi IP address assignment
Microsoft® .Net Framework 4.0 [31]	Parallel CPU processing. The total work load is split into 4 tasks which consist of Bandwidth optimization, GPS location transmission, 3G broadband data transmission and vehicle location trilateration.
Windows Management Instrumentation [32]	V2V WiFi IP address conflict reset.

Speech Recognition SDK 11.0 [33]	Based on Microsoft® OS Speech recognition
-----------------------------------------	-------------------------------------------

B. Hardware Setup

The system was tested on a 2.6GHz x86 Intel Centrino dual core mobile processor, with 4 Gigabytes of RAM and two NVidia® 8800M GTX GPU graphic card with Scale Link Interface (SLI) disabled. The 8800GTX consist of 768MB RAM, 6 stream multiprocessors, with each multiprocessor consist of 16 stream processors, which comes to the total of 96 stream processors. Each multiprocessor has 8192 of 32-bit registers; each multiprocessor can have up to 768 threads. Threads are partitioned into thread blocks which can consist of up to a maximum of 512 threads, and can be furthered be divided into warps of 32 threads.

The following hardware devices in Table VIII used in this work are:

TABLE VIII. HARDWARE SETUP

Device	Function	Methodology/Algorithm
OpenWRT Wi-Fi WRT54g router (802.11g) [34]	V2V communication for real time location, speed and time tracking	Based on Open source Wi-Fi stack and customized to fit specifically in this research.
OpenWRT Wi-Fi WRT54g wireless client adapter (802.11g) [34]	Client for V2V communication to OpenWRT router communication, up to maximum of 30 clients per router.	
ZigBee® Coordinator (802.15.4) [35]	ZigBee® client for V2V communication	Based on ember [36] ZigBee® stack. Based on RSSI for vehicle trilateration and Bandwidth optimization
ZigBee® Router (802.15.4) [35]	ZigBee® router to communicate with a ZigBee® coordinator	
GPS [37]	Vehicle location, time and date	NMEA 0183 GPS protocol to retrieve the current vehicle location.
3G broadband [38]	In an emergency situation, the device will send the vehicle location to database server.	standard 3G software protocol for transmitting and receiving from client to client , client to server and V2V communication.
LCD Touch Screen	Display vehicle location	-

C. Trial run

As a trial run, the system was tested in an outdoor environment on a local surface road. For safety reasons the vehicle maximum speed limit was set to 60kmh. For indoor testing, the GPS setting is deactivated. The vehicle to vehicle distance estimation is based on the ZigBee® trilateration and Wi Fi signal for distance judgment.

As shown in Fig. 5, the red spot shows the current vehicle location while the green spot indicate other vehicles within 100 meters radius. With this information, the system analyzes the potentially dangerous incoming vehicles and warns the driver

about vehicles at blind spot, hidden corners or idle vehicle on the middle of the road.



Figure 5. Map and vehicle info

Table IX shows the data transmitted from vehicle B to vehicle A during the test. However, the following data is used for debugging purposes only. It is hidden from the driver. The 1st column shows the date, 2nd column stores the time taken in 24 hours format, 3rd column is the unique ID from vehicle B which is "Client 1". The 4th column is the time difference in terms of speed in kmh. Negative value means the vehicle B is moving faster than vehicle A. The 5th and 6th column is the difference of longitude and latitude value between vehicle B and vehicle A. The negative value of the longitude shows the vehicle B is moving towards the right side. The last column, latitude with a positive value indicates that the vehicle B is front of vehicle A.

TABLE IX. VEHICLE TRACKING SYSTEM

05-04-12, Client1, 11:35:29, -5.092995, 0.00017, -0.00006
05-04-12, Client1, 11:35:29, -5.092995, 0.00017, -0.00006
05-04-12, Client1, 11:35:30, -3.223, 0.00019, -0.00006
05-04-12, Client1, 11:35:30, -3.223, 0.00019, -0.00006
05-04-12, Client1, 11:35:31, -0.5550003, 0.00020, -0.00006
05-04-12, Client1, 11:35:31, -0.5550003, 0.00020, -0.00006

Table X and Table XI are the RSSI value on a field test within line of sight. The error rate for Wi Fi Packet Error rate is within 10%. A 5dBi and 2dBi omnidirectional antenna is used for Wi Fi and ZigBee® devices. The test results have shown that the trilateration range estimation for ZigBee® is within ±3 meters tolerance.

TABLE X. WIFI RSSI*

Wi Fi (meters)	Transfer rate	RSSI (dB) - OpenWRT Router with 5dBi omnidirectional antenna	Packet Error Rate(PER)
10	≤54Mbps	-65dBm	<10%
50	≤48Mbps	-67dBm	
80	≤36Mbps	-70dBm	
100	≤12Mbps	-80dBm	

*Maximum Wi Fi client to router, Wi Fi router to router enumeration = 5 second

TABLE XI. ZIGBEE® RSSI*

ZigBee® (meters)	RSSI (dB) – with 2dBi omnidirectional Antenna	Trilateration error Tolerance
10	-40dBm	±3 meters
50	-58dBm	
80	-72dBm	
100	-83dBm	

*Maximum ZigBee® Node to node enumeration = 30ms

Fig. 6 and 7 shows the data transfer rate against distance for Wi Fi and RSSI against distance for ZigBee® signal respectively. The experimental results were obtained within line of sight of two vehicles on an express way during the test drive. The Wi Fi transfer rate from Vehicle 1 Wi Fi client to Vehicle 2 Wi Fi router is directly proportional to the distance. As for the ZigBee® router on vehicle 1 and 2, the data transfer rate remains consistent at 250Kbps as wireless signal strength directly proportional to the distance.

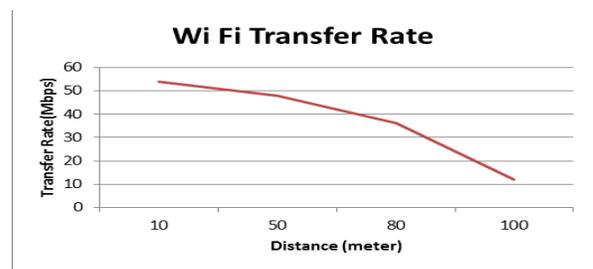


Figure 6. Wi Fi distance (meter) against transfer rate (Mbps)

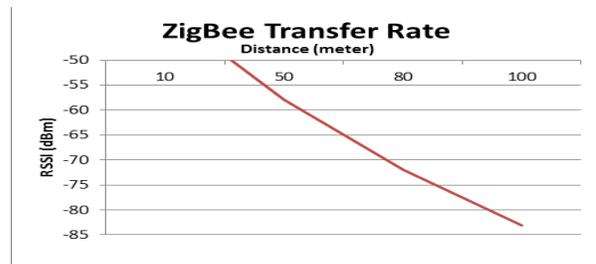


Figure 7. ZigBee® Distance (meter) against RSSI (dBm)

Table XII shows the results for the time taken for vehicle A to search for vehicle B within 100 meter radius. The total time taken to detect the vehicle in normal road condition has been reduced tremendously from 4.6 sec to 426 millisecond with the used of parallel CPU and GPU processing. In term of speed, it has significantly improved approximately 10 times on the system performance.

D. On road conditions

1) Scenario 1 Express way and road tunnel condition

Under normal traffic conditions, vehicles are communicating in a star topology protocol. There are two different wireless connections on each vehicle, Wi Fi and ZigBee® signal.

TABLE XII. PARALLEL CPU AND GPU ACCELERATION

Condition	Without Parallel CPU and GPU(millisecond)	With Parallel CPU and GPU (millisecond)
Cornering	≤4645	≤426
Tail gating	≤1787	≤289
Break Down at middle of road	≤1812	≤335
Lane Change	≤1884	≤383
Blind Spot	≤1921	≤399
Speed Track	≤2779	≤438

While waiting for the Wi Fi client to link up with other vehicles, ZigBee® coordinator and ZigBee® router have established the connection and estimated the trilateration distance between vehicle to vehicle and transmit the location through ZigBee® from vehicle to vehicle. At the same time, GPS location is transmitted through Wi Fi signal. If the vehicle is in a tunnel, the GPS will not be functional. Thus, the distance estimation will rely solely on ZigBee® RSSI from one vehicle to another.

Fig. 8 shows how the communication of Vehicle 1 to 5 is established on normal express way. Each vehicle is linked up to 30 vehicles either on a single hop Wi Fi router to Wi Fi client communication or Wi Fi router to Wi Fi router multi hop communication.

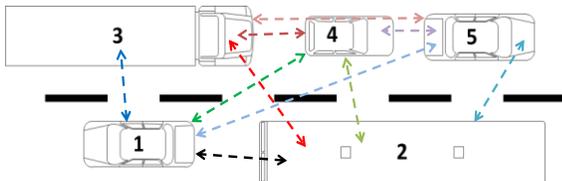


Figure 8. V2V - Normal condition

Table XIII shows how V2V communicate with each other on a road. From Fig. 8, Vehicle ID=1 communicates with the other vehicle ID=2, 3, 4 and 5, vehicle ID=2 communicates with vehicle ID=1, 3, 4 and 5 in a star topology. Similarly for vehicle ID=3, Vehicle ID = 4 and Vehicle ID = 5.

TABLE XIII. V2V COMMUNICATION

Vehicle ID	Vehicle Communication
1	2,3,4,5
2	1,3,4,5
3	1,2,4,5
4	1,2,3,5
5	1,2,3,4

2) Scenario 2 Traffic jam condition (ZigBee® off)

In this scenario, the system will pin point the vehicle current location using GPS solely. The GPS location from one vehicle to another is transmitted through Wi Fi until it reaches 100m radius or maximum of 30 vehicles (whichever comes first) on a single hop transmission. Although the Wi Fi takes a few seconds to establish a connection with the client, it is still

reasonable for the driver to wait for the time delay in a traffic congested area.

3) Scenario 3 Indoor car park condition (GPS off)

The 3rd scenario is in an indoor car park as shown in Fig. 9. ZigBee® routers and ZigBee® coordinators from vehicle 2 to 7 remained functioning even though the vehicle engine is switch off; the RSSI data is transmitted on every second interval. The car park is normally divided into different sections. Assume there are 10 vehicles parked on both left and right row. As vehicle 1 moved towards the direction of vehicle 2 and 5, the RSSI from vehicle 2 to 7 is received by the ZigBee® router of vehicle 1. The estimated error is ±3 meters, which is approximately 2 vehicle space.

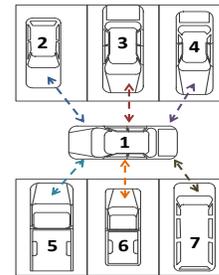


Figure 9. V2V - Indoor car park

VI. CONCLUSIONS

This paper has presented an efficient V2V communication implementation for driver safety by implementing the following methods: 1) Using a variety of real time V2V wireless communication protocols in different circumstances that optimizes performance, and 2) utilizing parallel CPU and GPU processing to accommodate real time processing of massive amounts of data. The methods implemented here are an improvement on past solutions due to the lower cost and lower power consumption of the proposed system without sacrificing performance.

The results in Table XII have shown that, by using parallel CPU and GPU, the time taken can be decreased up to 10 times of normal CPU alone implementations without parallel processing. However the results can further be improved when better hardware CPU and GPU options (in terms of performance per cost) become available. In comparison to other related work in the domain our results in Table XIV shows that the techniques implemented here are lower costs using lower power but with higher or similar speed performance.

Looking forward, the next objective is to port the current system to a mobile device such as a multi core OS hand phone. This porting will further reduce the cost and at the same no additional processing hardware is required other than the mobile phone. The total cost can further be reduced and the design can have much lower power consumption in order to achieve the same performance as the system discussed here.

TABLE XIV. COST AND PERFORMANCE COMPARISON

Criteria	Results for this research	GPS, GPRS/GSM system [16] [17] [18] [19] et. al	Radar/Ladar, camera, radio based system [20] [21] [22] [23] et. al	WAVE/RSU system [24] [25] [26] [27] et. al
Total cost	≤USD950	≤USD400	≤USD10K	≤ USD14K
Power consumption	≤150W ^a ≤ 1W for ZigBee® transceiver	≤50W ^a	≤150W ^a	Depending on RSU [44]
Accuracy	±3 meters	±15 meters	±1 meter	±1 meter
Speed	≤ 0.5 second	≤5 second	≤ 50 millisecond Radar [43] response time and ≤ 2 second total time ^b	≤ 2 second ^b

a. Whole System running at full load

b. Depending on the system configuration

REFERENCES

[1] UN, "Road Safety Report," 2012. [Online]. Available: <http://www.un.org/ar/roadsafety/pdf/roadsafetyreport.pdf>.

[2] C. F. W. X. Yuliya Kopylova, "Accurate Accident Reconstruction in VANET," in Data and Applications Security and Privacy XXV, Springer Berlin Heidelberg, 2011, pp. 271-279.

[3] C. Z. D. T. Soyoung Park, "Reliable Traffic Information Propagation in Vehicular Ad hoc Networks," in Reliable Traffic Propagation, vol. 197, World Scientific, 2008, pp. 1-29.

[4] "Driving Hazards Message Propagation Using Road Side Infrastructure in VANETs," in Advances in Computer Science and Information Technology, Computer Science and Information Technology, Springer Berlin Heidelberg, 2012, pp. 1-8.

[5] S. M. M. Zaydoun Y Rawashdeh, "A novel algorithm to form stable clusters in vehicular ad hoc networks on highways," EURASIP Journal on Wireless Communications and Networking, pp. 1-13, 2012.

[6] T. D. L. Anna Maria Vegni, "Hybrid vehicular communications based on V2V-V2I protocol switching," International Journal Vehicle Information and Communication Systems, vol. 2, no. 3, pp. 213-231, 2011.

[7] M. F. A. H. K. K. H. S.-W. Y. Ali Tufail, "An Empirical Study to Analyze the Feasibility of WIFI for VANETs," in IEEE press, 2008.

[8] S. P. Vaishali D. Khairnar, "V2V communication survey - (Wireless technology)," International Journal of Computer Technology & Applications, vol. 3, no. 1, pp. 370-373, 2012.

[9] S. J. KLAMPÁR M., "Design of communication systems among vehicles," in Proceedings of the 18th Conference EEICT, 2012.

[10] J. G. J. P. B. V. C. G. E. O. J. A. V. Milan'es, "V2I-Based Architecture for Information Exchange among Vehicles," in IFAC Symposium on Intelligent Autonomous Vehicles, 2010.

[11] H. Hartenstein, in VANET - Vehicular Applications and Inter-Networking Technologies, JohnWiley & Sons Ltd., 2010, p. 4.

[12] J. A. E. Toutouh, "Performance analysis of optimized VANET protocols in real world tests," in Wireless Communications and Mobile Computing Conference (IWCMC), 4-8 July 2011.

[13] A. A. M. S. M. S. K. N. G. N. Q. a. A. S. Muhammad Asif khan, "A Survey on Architecture, Protocols, Challenges and," International journal of multidisciplinary sciences and engineering, vol. 3, no. 3, pp. 5-8, 2012.

[14] I. A. Z. S. K. M. S. M. S. A. J. R. M. A. Rahim, "A comparative study of mobile and vehicular Ad-Hoc Networks," International Journal of Recent Trends in Engineering, vol. 2, no. 4, pp. 195-197, 2009.

[15] B. Mughal, A. Wagan and H. Hasbullah, "Efficient congestion control in VANET for safety messaging," in Information Technology (ITSim), 2010 International Symposium, 2010.

[16] F. I. M. C. B. G. P. D. G. K. Giorgio Rusconi, "I-WAY, intelligent co-operative system for road safety," in IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 2007.

[17] M. A. Al-Khedher, "Hybrid GPS-GSM Localization of Automobile Tracking System," International Journal of Computer Science & Information Technology, vol. 3, no. 6, pp. 75-85, 2011.

[18] R. K. Neha Verma, "Efficient Data Delivery For Secured Communication in Vanet," Journal of Computer Engineering, vol. 2, no. 2, pp. 1-8, 2012.

[19] D. J. P. H. L. S. L. Z. Yong Li, "Revealing Contact Interval Patterns in Large Scale Urban vehicular ad hoc networks," in Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, New York, NY, USA, 2012.

[20] K. T. Y. K. S. M. K. K. H. Park S. J., "A novel signal processing technique for vehicle detection radar," in Proceedings on IEEE MTT-S International Microwave Symposium Digest, 2003.

[21] A. M. E.Coelingh, "Collision warning with full autobrake and pedestrian detection — a practical example of automatic emergency braking," in Proceedings of the 13th International Intelligent Transportation Systems (ITSC) IEEE Conference, 2010.

[22] M. L. X. Q. F. L. Q. L. Tingting Zhou, "Trajectory Generation Model-Based IMM Tracking for Safe Driving in Intersection Scenario," International Journal of Vehicular Technology, pp. 1-8, 2011.

[23] C. W. G. Caicai, "Ground moving target tracking with VS-IMM particle filter based on road information," in Proceedings of the IET International Radar Conference, 2011.

[24] O. K. H. H. Thomas Mangel, "5.9 GHz inter-vehicle communication at intersections: a validated non-line-of-sight pathloss and fading model," Journal on Wireless Communications and Networking, pp. 1-11, 2011.

[25] H. A. C. A. M. R. S. Claudia Campolo, "Impact of Urban Radio Obstructions on effectiveness of moving WAVE providers," in Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, 2011.

[26] A. M. Claudia Campolo, "Improving V2R connectivity to provides ITS applications in IEEE 802.11p/ WAVE VANET'S," in Telecommunications (ICT), 2011 18th International Conference, Ayia Napa, 2011.

[27] M. J. Annette Bohm, "Real-Time Communication Support for Cooperative, Infrastructure-Based Traffic Safety Applications," International Journal of Vehicular Technology, pp. 1-17, 2011.

[28] "CUDA," NVidia, [Online]. Available: http://www.nvidia.com/object/cuda_home_new.html.

[29] v. studio, "www.microsoft.com/visualstudio/," [Online]. Available: www.microsoft.com/visualstudio/.

[30] WCF. [Online]. Available: <http://wcf.codeplex.com/>.

[31] ".Net Framework," Microsoft, 2011. [Online]. Available: <http://www.microsoft.com/net/>.

[32] WMI. [Online]. Available: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa394582\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa394582(v=vs.85).aspx).

[33] "Windows SDK for Windows 7," Microsoft, 2011. [Online]. Available: <http://www.microsoft.com/download/en/details.aspx?id=8279>.

[34] "OpenWRT," 2011. [Online]. Available: <https://openwrt.org/>.

[35] "ZigBee," ZigBee Alliance, 2012. [Online]. Available: <http://www.zigbee.org/>.

[36] "Ember," ZigBee, 2012. [Online]. Available: http://www.ember.com/products_zigbee_software.html.

[37] "GPS 18x," Garmin, [Online]. Available: http://www8.garmin.com/manuals/GPS18x_TechnicalSpecifications.pdf NHBfbZm5ZXtJ13QNpbjGCMn7GEQ.

[38] "Broadband modem," HuaWei, 2012. [Online]. Available: <http://www.huawei.com/>.

[39] "Threads Building blocks," Intel, 2011. [Online]. Available: <http://threadingbuildingblocks.org/>.

- [40] H. S. a. A. C. BhavneetSidhu, "Emerging Wireless Standards - WiFi, ZigBee," in World Academy of Science, Engineering and Technology, 2007.
- [41] NVidia, "CUDA Memories," [Online]. Available: <http://courses.engr.illinois.edu/ece498/al/textbook/Chapter4-CudaMemoryModel.pdf>.
- [42] NVidia, "NVidia CUDA C Programming Guide," 2012. [Online]. Available: <http://docs.nvidia.com/cuda/index.html>.
- [43] "ESR," 2012. [Online]. Available: http://www.autonomoustuff.com/uploads/9/6/0/5/9605198/radar_comparison_chart_as.pdf.
- [44] "Volpe," 2012. [Online]. Available: <http://www.volpe.dot.gov/coi/att/work/safety-pilot.html>.