Alandalus University for Science and Technology
College of Engineering
Department of Information Technology

Tracking Service in Vehicular networks based on Cloud Computing

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Information Technology
College of Engineering
Alandalus University for Science and Technology

By

Salah Mohammed Abduljalil Ali

Supervisor
Associate Professor: Fekri M.Abduljalil, Ph.D.

2017

بسم الله الرحمن الرحيم

(وَقُلْ رَبِّ زِدْنِي عِلْماً)

صدق الله العلي العظيم

سورة طه: من الآية114

DECLARATION

I declare that the thesis entitled **"Tracking Service in Vehicular networks based on Cloud Computing"** *submitted by me for the degree of Master is the record of work carried out by me during the period from* ***23/04/2016*** *to* ***26/04/2017*** *under the guidance of* **Associate Professor Dr. Fekri M.Abduljalil** *and has not formed the basis for the award of any degree, diploma, associates hip, fellow-ship, titles in this or any other University or other institution of Higher learning.*

I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

Mr. Salah Mohammed Abduljalil Ali Date: July-2017

University of **Alandalus for Science and Technology**

2017

CERTIFICATE

CERTIFIED that the work incorporated in the thesis **"Tracking Service in Vehicular networks based on Cloud Computing"** Submitted by **Mr. Salah Mohammed Abduljalil Ali** was carried out by the candidate under my supervision/ guidance. Such material as has been obtained from other sources has been duly acknowledged in the thesis.

Place: Sana'a

Date: 2017

Associate Professor Dr. Fekri M.Abduljalil

Research Guide

ACKNOWLEDGEMENTS

I sincerely thank my advisor Dr. Fekri M.Abduljalil for giving me the opportunity to pursue my research under his guidance. It has been a great learning experience working with him. Special thanks to all my friends for their enthusiastic and generous support through my research study. Above all, I thank my parents and brothers who have been a constant source of encouragement and support for me throughout my journey in the Master's Degree program.

DEDICATION

This work is dedicated to my family and friends, especially my wonderful, beautiful, loving, and supportive wife Om Mohammed. I love you more than you will ever know. This is also dedicated to Mohammed, my new son – I hope that this work will inspire you and show that with hard work, dedication, and the support of loved ones anything is possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

3G	3rd Generation
CaaS	Cooperation As A Service
CC	Cloud Computing
CDMA	Code Division Multiple Access
DSRC	Dedicated Short Range Communications
ENaaS	ENtertainment As A Service
GIS	Geographic Information System
GPS	Global Positioning System
GSM	Global System For Mobile Communication
I2V	Infrastructure-To-Vehicle
INaaS	Information As A Service
LAISYC	Location Aware Intelligent System
LBA	Location-Based Application
LBAC	Location Based Application Client
LBS	Location Based Service
LBSaaS	Location Based Services As A Service
LBSC	Location Based Service Client
LUM	Location Update Message
LUP	Location Update Problem
MCC	Mobile Cloud Computing
NaaS	Network As A Service
NIC	Network Integrated Circuit
OBU	On Board Unit
pic-on-wheel	Picture On Wheel

R2V	Roadside-To-Vehicle
STaaS	STorage As A Service
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
TSaaS	Tracking Services As A Service
TSVCC	Tracking Services in Vehicular Cloud Computing
UDP	User Datagram Protocol
UTDOA	Uplink Time Difference Of Arrival
V2V	Vehicle-To-Vehicle
VANET	Vehicular Ad-hock Network
VCC	Vehicular Cloud Computing
VIN	Vehicle Identifier Number
Wi-Fi	Wireless Fidelity

ABSTRACT

Tracking Service in Vehicular networks based on Cloud Computing

By

Salah Mohammed Abduljalil Ali

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the

Master of **Information Technology**

Alandalus University for Science and Technology

2017

The increasing numbers of proposed services in the vehicular network contribute for building big intelligent systems such as smart cities and intelligent transportation systems. The main target is helping peoples on the road. Recently, many location-based systems are proposed in vehicular cloud computing as applications to do specific functions such as anti-theft car systems, vehicles monitoring systems and fleet management …etc. This type of applications depends on real-time vehicle's location to which we refer as Location-Based Applications (LBA) or Location-Based Services (LBS).

This thesis presents a novel Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) platform as a service, to support developing many location-based applications in VCC environment. This thesis also presents an optimized VCC architecture which is implemented in a testbed using .Net Frameworks to evaluate the performance of LBA. It is improved network bandwidth utilization and minimized storage capacity needs on location server. It decrease number of location update messages a vehicle should send to a location server. The proposed system is evaluated and the results are discussed.

Chapter 1

Introduction

CHAPTER 1 INTRODUCTION

This chapter introduces Vehicular Cloud Computing (VCC), smart vehicle, vehicle application, Location-Based Services (LBS), positioning technologies, problem formulation, motivations, research approach and contributions. The structure of this thesis is presented at the end of this chapter.

1.1 Vehicular Cloud Computing (VCC)

Every day, many vehicles spend hours in a parking garage or moving on streets. The parked vehicles are a vast unexploited resource which is currently simply lost. Proposed features of vehicle make vehicles the perfect candidates for nodes in a cloud computing network. As a result, the cloud computing paradigm has enabled the exploitation of excess computing power. The vast number of vehicles on streets, roadways and parking lots will be considered as under-utilized computational resources which can be used for providing public services. Some vehicle owners may agree to rent out excess on board resources, similar to the holders of huge computing and storage facilities who rent out their excess capacity and benefit economically. The travelers normally leave their vehicles in airport parking spaces while they are traveling. The airport management will power the vehicles' computing resources and allow for on demand access to this parking garage datacenter. Similarly, the drivers stuck in traffic congestion will agree donate their on board computing resources to help city traffic authorities run complex simulations designed to remove congestion by rescheduling the traffic lights of the city.

The Concept of Vehicular Cloud Computing (VCC) is solution shifted from Mobile Cloud Computing (MCC) for best exploitation of vehicular network resources. Concepts of VCC come from a combination of cloud computing advantages with unexploited vehicular networks resources. The VCC is new technology that has an unusual impact on traffic management, road safety and other application by on-demand model using of vehicular resources, such as computing, storage and internet for supporting customers with new services and applications [3][4].

1.1.1 Smart Vehicle

Newly Objects such as (Smart-Phone, Smart-Vehicle, Smart-Home, and Smart-City ….etc.) have become more smartly due to the combination of complex technologies operating together. Some modern countries like Japan [18], Canada [19], USA [16] and others have projects to improve utilities of the smart-vehicles for improving people's life in this world.

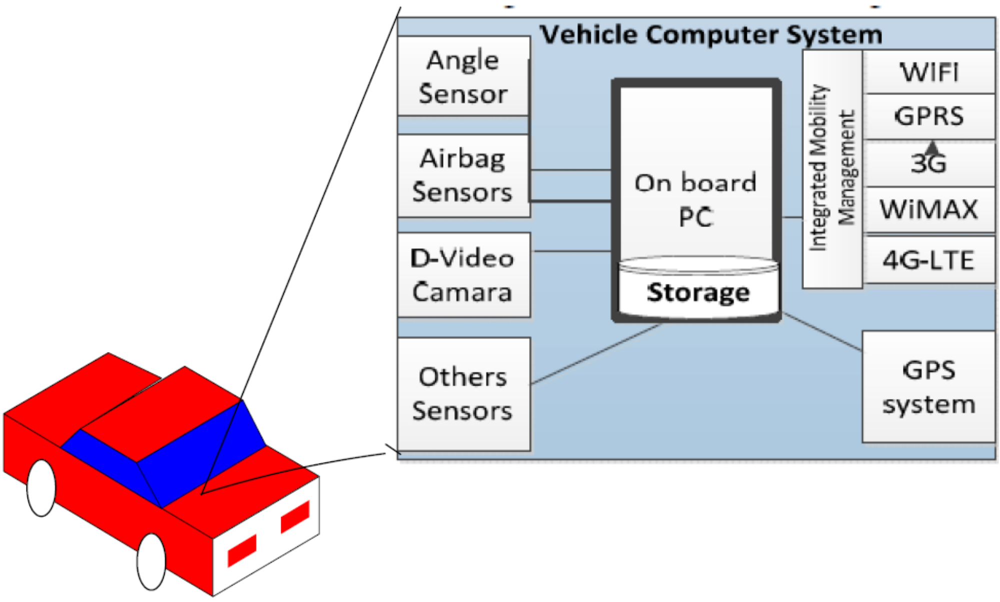

Figure 1-1 On Board Unit (OBU)

Recently, vehicular networking has become an unusual research area due to the following points:

- Vehicles are considered as computers on wheels due to them are carrying more resources On-Board Unit (OBU) such as communication systems, storage and sensors [2] as illustrated in Figure 1-1. On-Board

Unit (OBU) has many components consider as resources on vehicle can be used.

- Its specific applications such as efficient traffic management, road safety and infotainment [1].

1.1.1.1 Vehicle Applications

Vehicles have become very popular not only for their ability to transport the passengers, but also for their ability to perform general computational tasks that previously required expensive computers. Maybe one of the most popular features of modern smart vehicle is the ability to execute vehicle applications. Vehicle applications, or “apps,” are software products that are typically developed by a third-party that is not necessary have a direct relationship with the vehicle manufacturer (e.g., Toyota, Mazda, GMS and Ford) or operating system vendor (e.g., Google, Microsoft). Instead, the vehicle application is created by software engineers and then directly sold and distributed to the customer. As a result of an increasing of smart vehicles capabilities, the number of vehicle apps will be increased.

1.1.1.1.1 Location-Based Services/Applications (LBS/A)

Location-Based Services (LBS) have grown fast in the last decade due to pervasive adoption of GPS. Location-Based-Application (LBA) is a type of vehicular cloud computing applications that depends on real-time vehicles’ location. LBA systems are tracking vehicles’ location and maintaining an accurate up-to-date view of the entire vehicular network. Periodically, each vehicle updates its real-time location to server, but this scheme has some challenges, storage, network bandwidth overhead and server overhead which is called Location Update Problem (LUP) [5].

Many strategies have been proposed as solutions for location update problem. The main goal was minimizing the number of the update location messages should be send to location servers [5] [25].

In vehicular cloud computing, the real-time vehicle's location is a cornerstone in several proposed location-based service such as Pic-on-wheel as a service [11], Vehicle Witnesses as a Service [6], Video capture [2], Tracking systems [8], and Traffic analysis [12].

In the above mentioned location-based services in vehicular cloud computing, the impact of position updates (a function of the frequency of the position data being sent to a server) on vehicle network bandwidth's consuming is not directly considered while the same single vehicle has more than one LBS.

In VCC environment, we face a Location Update Problem (LUP). What if a single vehicle has more than one LBA? Normally, the LUP will be doubled by the number of LBAs installed on the single vehicle. The proposed LBAs in vehicular cloud computing needs a scalable and efficient platform which make the others third-party develop an innovative location-based service while tracking real-time vehicles' location.

In this thesis, we propose a Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) platform which produces a set of web services consumed as a service while TSVCC platform is tracking real-time vehicles' location in VCC environment. It supports the development of many location-based applications.

TSVCC contributes the improvement of network bandwidth consuming and decreases the number of location update messages should be sent. It treats impact of increasing number of LBAs in same single vehicle.

1.2 Positioning Technologies

Some authors consider vehicle as computer on wheels [11], but the difference between vehicles and desktop computers is that vehicles forever change geographic location, unlike desktop computers, which are stay at a single physical location for months or years. Even laptops do not have the level of mobility that vehicles offer. Laptops can be moved from one place to another, but typically they are in operation for only several hours at a time and then shut down before being moved. In contrast, typically vehicles remain on street during the entire day and can be used when vehicle is in drive away or in parking.

Early, deployments of positioning technologies for commercial purposes become known as location-based services (LBS) which are a general class of services that provide consumers, its depends on real-time vehicle's location .

Global Positioning System (GPS)-based solutions are by far the most accurate, with an estimated 3-5 meters of positional accuracy under ideal conditions [22]. Since this level of accuracy is also sufficient to provide commercial services such as real-time driving directions to VCC users.

It is essential to note the word Location-Based Service (LBS) since the terms location-based service and Location Based Application (LBA) are sometimes used interchangeably in conversation.

It is essential to note the word vehicle positioning since the terms vehicle positioning and vehicle's location are sometimes used interchangeably in conversation, but they are really two different things. Vehicle positioning refers to determining the position of the vehicle. Vehicle's location refers to the location estimate derived from the vehicle positioning operation. The following sub section describes positioning technologies which can be used in location-based applications.

1.2.1 Global Positioning System (GPS)

Today most vehicles supports with satellite positioning system is Global Positioning System (GPS). In April 1995 the whole system was containing 24 operational satellites in an 11000 nautical mile orbit [24]. This feature can be used in vehicular cloud computing to develop location-based applications.

1.2.1.1 How GPS Work

GPS use satellite data to calculate an accurate position on the earth. These calculations relate the vehicle position execute within mille-seconds. Its many types from GPS that its work in a similar manner. The most significant difference between GPS receivers is the number of satellites they can simultaneously communicate with. Most receivers are described as 12 channels meaning they can communicate with 12 satellites. Older models may be 8 or even 5 channels with more modern receivers capable of communicating with 14 – 20 satellites.

GPS data do not contain positional data. The positional data calculated by the receiver on the ground is a calculated position based on range-finding triangulation [40]. GPS positioning is achieved by measuring the time taken for a GPS data to reach a receiver. Almost one million times a second the satellite transmits a one or a zero in a complex string of digits that appears random. In actuality this code is not random and repeats every 266 days. The receiver knows that the portion of the signal received from the satellite matches exactly with a portion it generated a set number of seconds ago. When the receiver has determined this time, the distance to the satellite can be calculated using simple trigonometry where: Distance to the *satellite = speed x* $(t_r - t_{to})$ (where speed is c, the speed of light, in a vacuum (299792.5 X 10^3 ms^{-1}). t_{to} is the time at the origin and t_r is the time at the receiver). This principle is demonstrated in the Figure 1-2 briefly.

(a) From one measurement we know the receiver can be anywhere at a uniform distance from the satellite with a radius equal to $r = c X (t_r - t_{to})$. This defines the outer surface of a sphere of radius r.

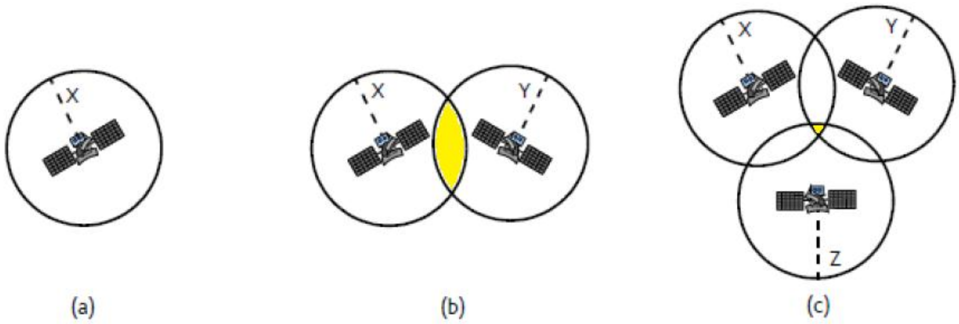

Figure 1-2 GPS system

(b) From two measurements we know the receiver must be anywhere on the line of the outer edge of a circle of intersection between the two spheres shown as a shaded ellipse in figure 1-2 (b)

(c) A third measurement reduces this to the intersection of a plane with the circle. This reduces the possible location to one point on the Earth's surface.

1.2.2 Assisted-GPS (A-GPS)

In order to achieve high location accuracy at reasonable cost. The Assisted-GPS (A-GPS) technology overcomes the drawbacks of the conventional GPS solution. The assistance to the mobile device trying to determine its own location comes from the network over the air-interface, and this distributed approach exceed better performance level than conventional GPS exceed. This technology works in the mobile network, using its own GPS receivers, as well as an estimate of the mobile's location close to cell/sector, can predict with great accuracy the GPS signal and transfer that information to the mobile. The A-GPS method can be extremely accurate, ranging from 1 to 10 meters [22].

1.2.3 Differential GPS (D-GPS)

Differential correction methods are used to enhance the quality of position data gathered using global positioning system (GPS) receivers. Differential correction can be applied in real-time directly in the field or during post processing of data. Although both methods are based on the same underlying principles, each accesses different data sources and achieves different levels of accuracy. Combining both methods provides flexibility during data collection and improves data integrity. The idea of differential GPS (D-GPS) is that any two receivers that are relatively close together will test similar atmospheric errors. D-GPS requires that a GPS receiver be set up on an exactly known location. This GPS receiver is the base or reference station. The base station receiver calculates its position based on satellite signals and compares this location to the known location. The difference is applied to the GPS data recorded by the second GPS receiver, which is known as the portable receiver. The corrected information can be applied to data from the portable receiver in real-time in the field using radio signals or through post processing after data capture using special processing software [40].

1.3 Problem Formulation

Recently, VCC is open area to produce new kind of services that help human to get better and easy life on this world. LBS in vehicular cloud computing is interest but location-based applications in VCC face with location update problem, many researches provide solution to location update problem which take in account each a single vehicle contain a single LBA but this not enough in real-life. We can observe that, in the current system of smart phone, a single phone has many LBAs. VCC has been shifted from Mobile Cloud Computing (MCC), so in real life a single vehicle may be containing more than one of LBA on the same vehicle. So the traffic in vehicular networks overhead can be unmanageable when thousands of vehicles must update their locations within short time. It is proposed to develop efficient Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) platform as a service for supporting third-party LBAs in vehicular networks and handling impact of increasing number of LBAs on a single vehicle. It should have the following features:

- Less overhead bandwidth.
- Less number of location update massages(less storage capacity need).
- Efficient support third-party LBAs.

1.4 Motivations:

In vehicular cloud computing, the location update of vehicles is a fundamental capability in all types of Location-Based Applications (LBA), such as Traffic Management, Congestion Avoidance on Road, Cargo Tracking, Child Care, and Location-Based Advertisement and Entertainment.

1.5 Research Approach

This research addresses the VCC location update problem in LBA systems when single vehicle has many LBAs. My approach is to optimize the VCC Architecture in order to support third-party LBAs developer. In this thesis a new approach is proposed to reduce the number of location update messages that vehicles should be send to servers.

1.6 Contributions

This thesis presents the proposed novel Tracking Service in Vehicular Cloud Computing (TSVCC) platform that meets the needs of intelligent real-time vehicle's location applications and it is fully useable by third-party vehicular application developers. The proposed system architecture is implemented. Each component in the proposed system architecture has been implemented and tested as part of our research to demonstrate that our proposed system is fully implementable.

1.7 Structure of Thesis

The remainder of this thesis is organized as follows: Chapter 2 provides a detailed background and related works of our proposed. We start with overview Vehicular Cloud Computing (VCC) then Vehicular Cloud Computing Architecture after that, LBS in Vehicular Cloud Computing with the major limitations of LBS, it discussed in literature and related works then we illustrated the location-based services data flow. Chapter 3 presents the proposed Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) architecture that is the main subject of this thesis, and Chapter 4 presents an implementation of proposed system and evaluation. Chapter 5 concludes the thesis with an overview of the contributions and future research directions related to TSVCC.

Chapter 2

Background and Related Works

CHAPTER 2 BACKGROUND AND RELATED WORKS

The literature reviewed in this chapter includes overview on Vehicular Cloud Computing (VCC), Vehicular Cloud Computing Architecture, the potential Location-Based Services (LBS) and its major limitations, which can severely overhead network's bandwidth. It reviews of the proposed LBS on vehicular cloud computing, it highlights on location update problems and existing Strategies.

2.1 Vehicular Cloud Computing (VCC)

Normally, many vehicles spend hours in a parking garage or moving on street. The parked vehicles are a vast unexploited resource, which is currently simply lost. Proposed features of vehicle make vehicles the perfect candidates for nodes in a cloud computing network. As a result, the cloud computing paradigm has enabled the exploitation of excess computing power.

The vast number of vehicles on streets, roadways and parking lots will be considered as under-utilized computational resources, which can be used for providing public services. Some vehicle owners may agree to rent out excess on board resources, similar to the holders of huge computing and storage facilities who rent out their excess capacity and benefit economically.

The travelers normally leave their vehicles in airport parking spaces while they are traveling. The airport management will power the vehicles computing resources and allow for on demand access to this parking garage datacenter. Similarly, the drivers stuck in traffic congestion will agree donate their on board computing resources to help city traffic authorities run complex simulations designed to remove congestion by rescheduling the traffic lights of the city [3-4].

Many location-based applications proposed in vehicular cloud computing to enhance people lives, so this thesis contribute to improve performance of location-based applications.

The VCC is new technology that has an unusual impact on traffic management, road safety and other application by on-demand model using of vehicular resources, such as computing, storage and internet for supporting customers by new services and applications [3][4].As a results to previous motivation we introduce contribution in this area.

2.1.1 Vehicular Cloud Computing Architecture

In [3][4], authors presented state-of-art survey of vehicular cloud computing, they classified of VCC applications and services, they proposed VCC architecture which it used as a VCC reference model for our work.

As shown in figure 2-1, the vehicular cloud computing architecture consists from three primary layers:

Inside-vehicle layer: it consists of multiple components that are responsible for collecting information by On Board Unit (OBU) sensors. The information collated via sensors should be sent to the cloud for storage or for use as input for various software applications in the application layer. The inside-vehicle layer has vehicle operation system, multi task operation system. It can run many applications at same time on a single vehicle. These applications execute instructions (for example, take picture [11], send location [8], take video [2] and etc.) come from cloud application layer.

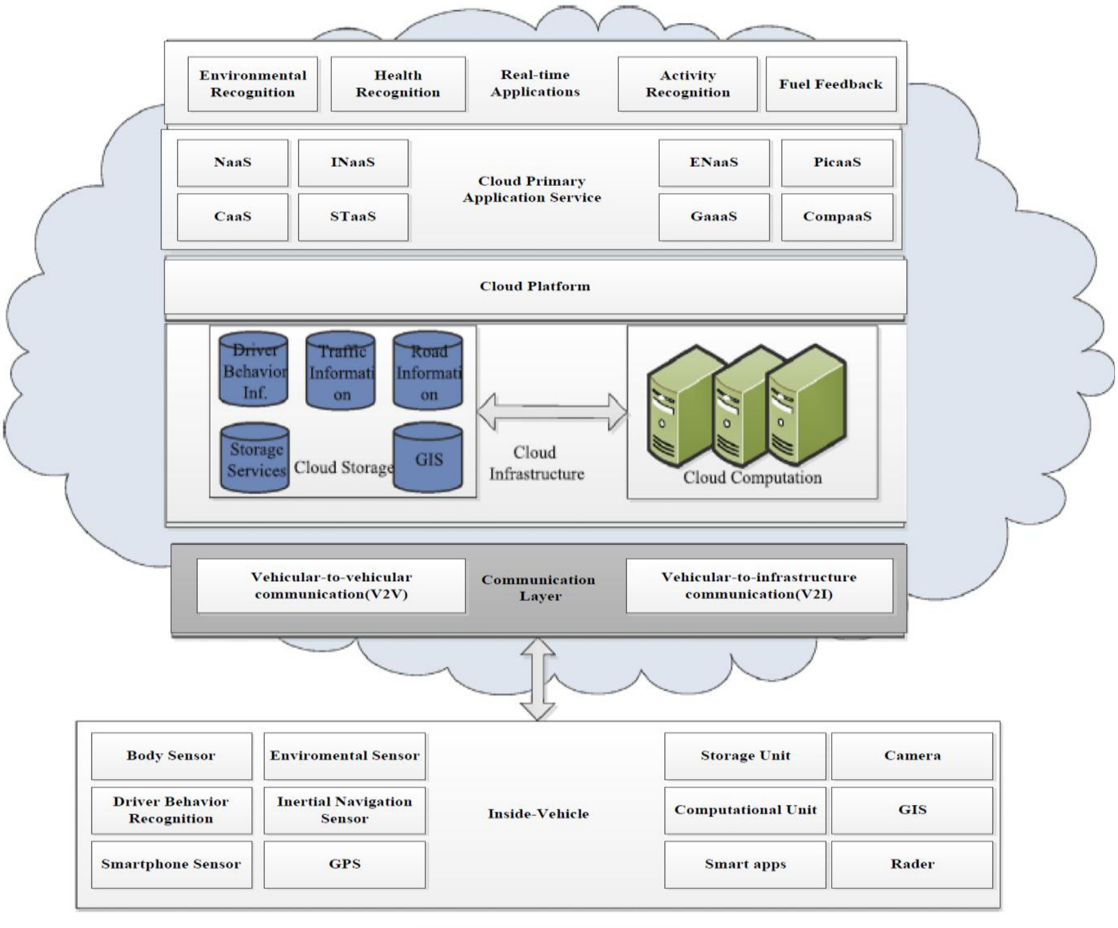

Figure 2-1 Vehicular Cloud Computing Architecture [3]

Communication layer: it is the middle layer, which responsible to exchange data between inside-vehicle layer and cloud layer. Vehicles may acquire information and services through the V2V (Vehicle-to-Vehicle) and I2V (Infrastructure-to-Vehicle) or R2V (Roadside-to-Vehicle) communications. The V2V communication is based on the Dedicated Short Range Communications (DSRC) technology [13]; while the I2V communication is based on GPRS/3G, Wi-Fi or WiMAX, WAVE 802.11p.

Cloud layer: it *is* the last layer of the VCC architecture. It can compute the massive and complex computations in minimal time. The cloud layer consists of three internal layers: **application**, **cloud platform**, and **cloud infrastructure**. In the application layer, various applications and services are considered as real-time services or cloud primary services. Real-time services are accessible remotely by drivers, such as fuel feedback, human activity recognition, health recognition and environmental recognition. Human activity recognition is used for an automated analysis (or interpretation) of ongoing events and their context of video data [3-4]. In the primary services, several services are deployed, such as Network as a Service (NaaS), Storage as a Service (STaaS), Cooperation as a Service (CaaS), Information as a Service (INaaS), and Entertainment as a Service (ENaaS).

The cloud infrastructure consists of two parties: cloud storage and cloud computation. The data gathered by the inside-vehicle layer will be stored in the cloud storage (geographic information system (GIS), road traffic control device). A storage system is based on the type of applications. The computation part used to calculate the computational task which provides faster performance.

2.1.2 LBS in Vehicular Cloud Computing

This section reviews the proposed applications of location-based services in vehicular cloud computing, location update problems and existing strategies.

Location-Based Services (LBS) have seen fast growth in the last decade due to pervasive adoption of GPS. Location-Based-Application (LBA) is a type of vehicular cloud computing applications that depends on real-time vehicles' location. In vehicular cloud computing, the real-time vehicle's location is a cornerstone in several proposed location-based service such as Pic-on-wheel as a service [11], Vehicle Witnesses as a service [6], Video capture [2], Tracking systems [8], and Traffic analysis [12].

In the above mentioned location-based services in vehicular cloud computing, the impact of position updates (a function of the frequency of the position data being sent to a server) on vehicle network bandwidth's consuming is not directly considered while the same single vehicle has more than one LBS.

LBA systems are tracking vehicles' location and maintaining an accurate up-to-date view of the entire vehicular network. Periodically, each a vehicle updates its real-time location to server, but this scheme has some challenges, storage, network bandwidth overhead and server overhead which is called Location Update Problem (LUP) [5].

Many strategies have been proposed as solutions for location update problem. The main goal was minimizing the number of the update location messages should be send to location servers [5] [25].

In VCC environment, we face a Location Update Problem (LUP). What if a single vehicle has more than one LBA? Normally, the LUP will be doubled by the number of LBAs installed on the single vehicle. The proposed LBAs in vehicular cloud computing needs a scalable and efficient platform which make the others third-party develop an innovative location-based service while tracking real-time vehicles' location.

In this thesis, we propose a Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) platform which produces a set of web services consumed as a service while TSVCC platform is tracking real-time vehicles' location in VCC environment. It support*s* the development of many location-based applications. TSVCC contributes the improvement of network bandwidth consuming and decreases the number of location update messages should be sent. It treats impact of increasing number of LBAs in same single vehicle.

2.1.2.1 Potential LBS Application In Vehicular Cloud Computing

Table 2-1potential LBS application in vehicular cloud computing

No	LBS Name
1	Real-time Traffic Condition Analysis and Broadcast
2	Real-time Car Navigation
3	Video Surveillance
4	LBS Commercial Advertisement
5	Vehicular Mobile Social Networking
6	In-Vehicle Multimedia Entertainment
7	Inter-Vehicle Video and Audio Communications
8	Remote Vehicle Diagnosis

According to [1], with powerful cloud computing, cloud-based vehicular networks can support many novel applications. The potential LBS applications in vehicular cloud computing are shown in Table 2-1. The impact of position updates (a function of the frequency of the data being sent to a server) on vehicle bandwidth consume when same single vehicle has more than one LBS application is not directly considered in the above mentioned LBS in vehicular cloud computing. We will review some location-based services proposed in vehicular cloud computing which are implemented.

2.1.2.1.1 Pics-On-Wheels Service

According to [11], it proposes a new vehicular cloud service. It is on demand utility of the available resource of vehicles to capture image and send it to customer. The Pics-on-Wheels service can also be used to retrieve pictures taken spontaneously by cars as part of background ambient surveillance. After several years in future will see this service consider as a vehicle application.

The Pics-on-Wheels Service selects a group of vehicles to take photo shots of a given urban landscape within a given timeframe as requested by a customer. To participate in this Service, vehicles register to the centralized cloud manager as hardware provider. They also upload their own GPS location periodically to the cloud manager. This scheme causes extra bandwidth consume and cause network congestion [5].

The pic-on-wheels service is implemented, but impact of the increasing number of location-based services inside single vehicle is not considered. Pic-on-wheels system is shown in figure 2-2.

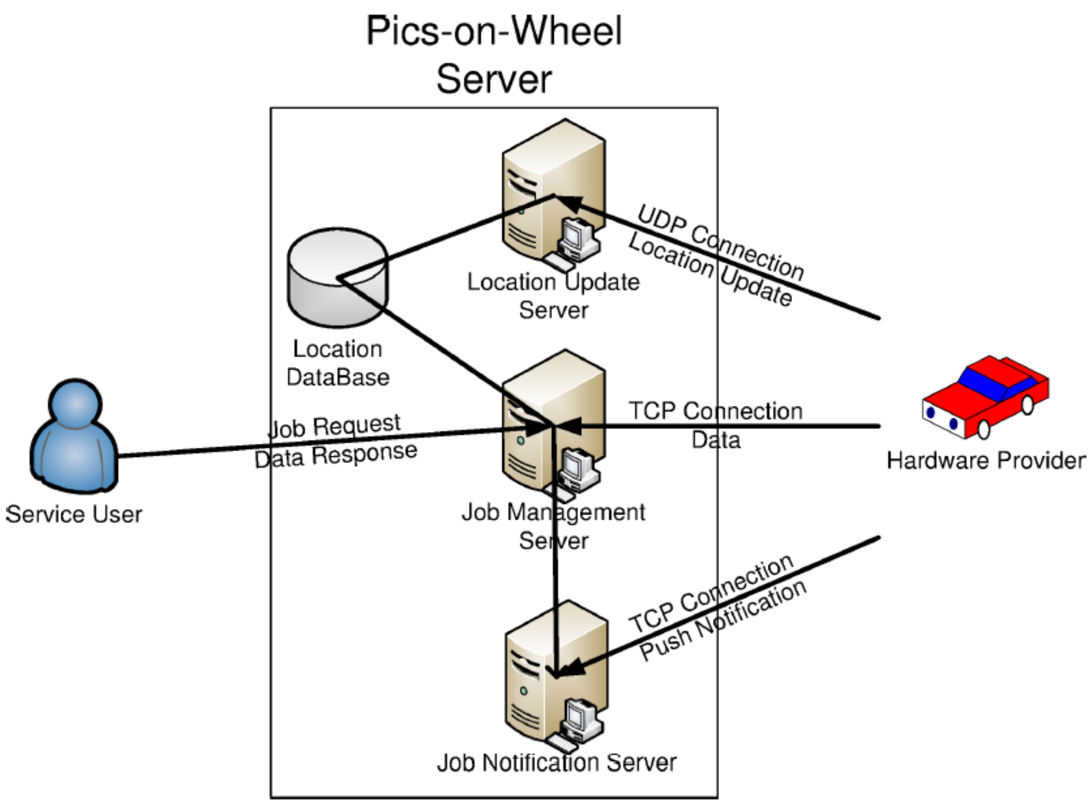

Figure 2-2 Pics-on-Wheels Server [11]

2.1.2.1.2 Video and Data Capture as a Service

In [2], it proposed a novel real-time video and data capture of vehicular accident management in intelligent transportation systems. It is on demand utility of the available resource of vehicles to video capture and send it to customer, accident management in intelligent transportation systems need to know where and when vehicles get accident, so that author proposed solution, each vehicle sending location update message periodically to location server. This scheme causes extra bandwidth consume and cause network congestion [5]. Impact the increasing number of location-based services inside single vehicle not considered.

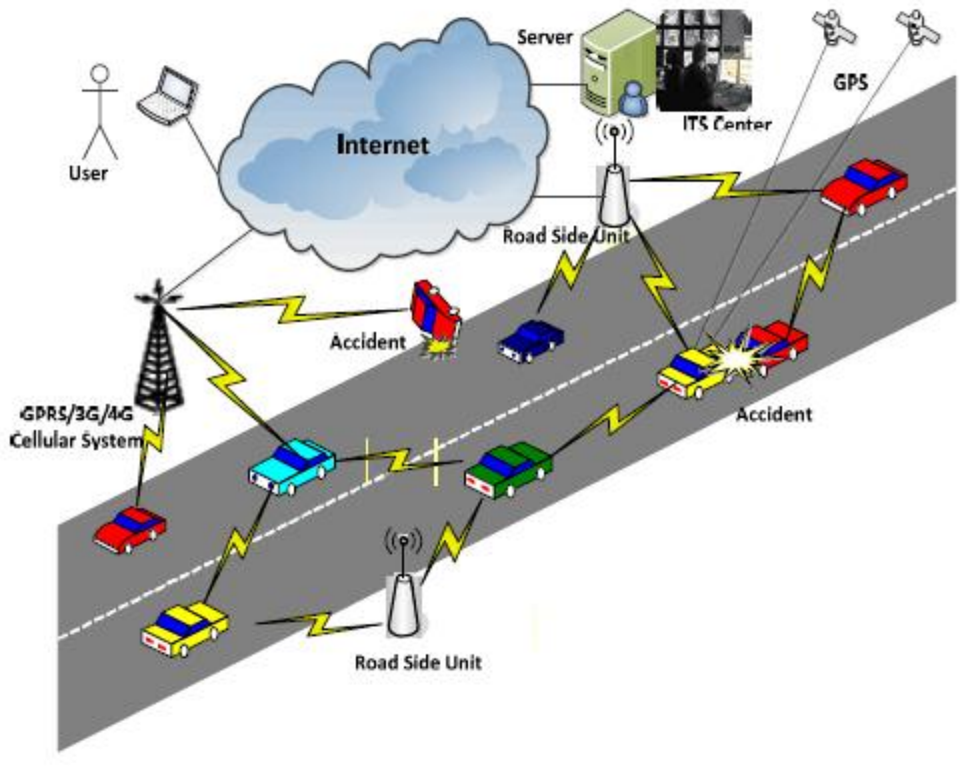

Figure 2-3 system model and architecture for real-time video and data capture [2]

Figure 2-3 shows system model and architecture for real-time video and data capture. The vehicle captures real-time video and data and saves it in its secondary storage and sends it to server. The server receives captured video and data and saves them in its storage. The user can access real-time video and data through request server. The system model can be used by traffic police or ITS Centre or any interested company such as car rent company.

2.1.2.2 Location Update Problems and Existing Strategies

A number of positioning systems are made publicly available for tracking the location update of vehicles moving on the road network, such as Google's Latitude and Skyhook wireless Wi-Fi positioning system [24]. Frequent location updates enable the location server to keep track of vehicles clients' current locations and ensure the accuracy of the location query results. The algorithm that mobile client's employ to determine when and where to update their locations is often referred to as the location update strategy. We review five location update strategies in following subsections:

2.1.2.2.1 Periodic Update Strategy

A periodic update strategy is the simplest time-based location update strategy, in which the location server maintains the location update for each vehicle at a fixed time interval. This update strategy implies that vehicles are treated as stationary between updates [5] [25].

2.1.2.2.2 Point-Based Update Strategy

This approach uses the distance based scheme and the server only record an update when the mobile client travels more than a delta threshold away in distance from the location of last update. The number of location updates per unit time will depend upon the speed of the vehicle [5].

2.1.2.2.3 Vector-Based Update Strategy

A vector based update strategy uses the speed vector of the vehicle to make a simple prediction about its location. An update is only sent when the current location of the vehicle deviates from its predicted location by an amount that is larger than a system-defined delta distance threshold. This strategy treats the velocity vector of the client as constant between updates [5].

2.1.2.2.4 Segment Based Update Strategy.

A segment based update strategy utilizes the underlying road network to limit the number of updates. Vehicles are assumed to move at a constant speed on their current road segment. An update is sent when the distance between the current and the predicted location is larger than a system-defined delta threshold. We assume that vehicles change their velocities at the end of each segment, i.e., the vehicle is assumed to have stopped at the segment end node and can change its movement speed and direction and move forward accordingly. Thus an update will be sent when the mobile client departs from a segment end node by delta distance. [25]

2.1.2.2.5 Awareness Based Update Strategy

In [5] it is proposed a query-aware location update framework for Mobile Clients. For decrease number location update message send to location server. All the mobile clients must aware of query. In this scheme assume that mobile clients and the location server have a local copy of the same road network database .the focused was how mobile clients send little number of the location update messages to location server to reduce bandwidth consume and server overhead, it take in account each a single mobile client has a single LBA. But the single vehicle in the VCC may carry more than one LBA.

2.2 Location-Aware Architecture Supporting Intelligent Real-Time Mobile Applications

In [23], A Location-Aware Architecture Supporting Intelligent Real-Time Mobile Applications is proposed. It focuses on improving phone battery life but this limitation is not point in vehicular network. It presented LAISYC, a modular location-aware architecture for intelligent real-time mobile applications that is fully-implementable by third party mobile app developers. It does not address impact of the increasing number of LBS in single vehicle. As shown in figure 2-4, The LAISYC architecture consists of software on the mobile device and web application server, with a database server holding persistent server-side data. Our approached is shifted from this work. This work considers the base of our idea.

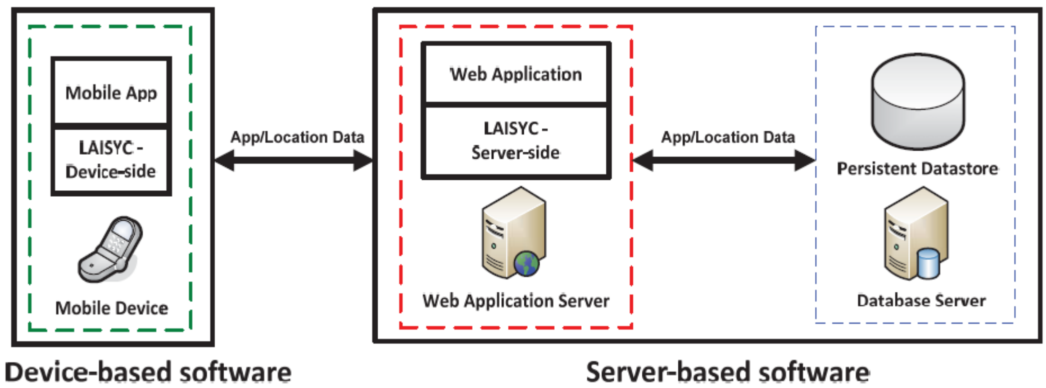

Figure 2-4 LAISYC architecture [25]

2.3 Data Flow in Location Based Service of VCC

In order to illustrate behavior of LBA in VCC, we design the LBA dataflow diagram that is shown in figure 2-5. It highlights on common characteristics of location-based services which are mentioned previously.

Location-Based-Application (LBA) /Location-Based-Service (LBS) is a type of vehicular cloud computing applications that depends on real-time vehicle's location. LBA systems are tracking vehicles' location and maintaining an accurate up-to-date view of the entire vehicular network.

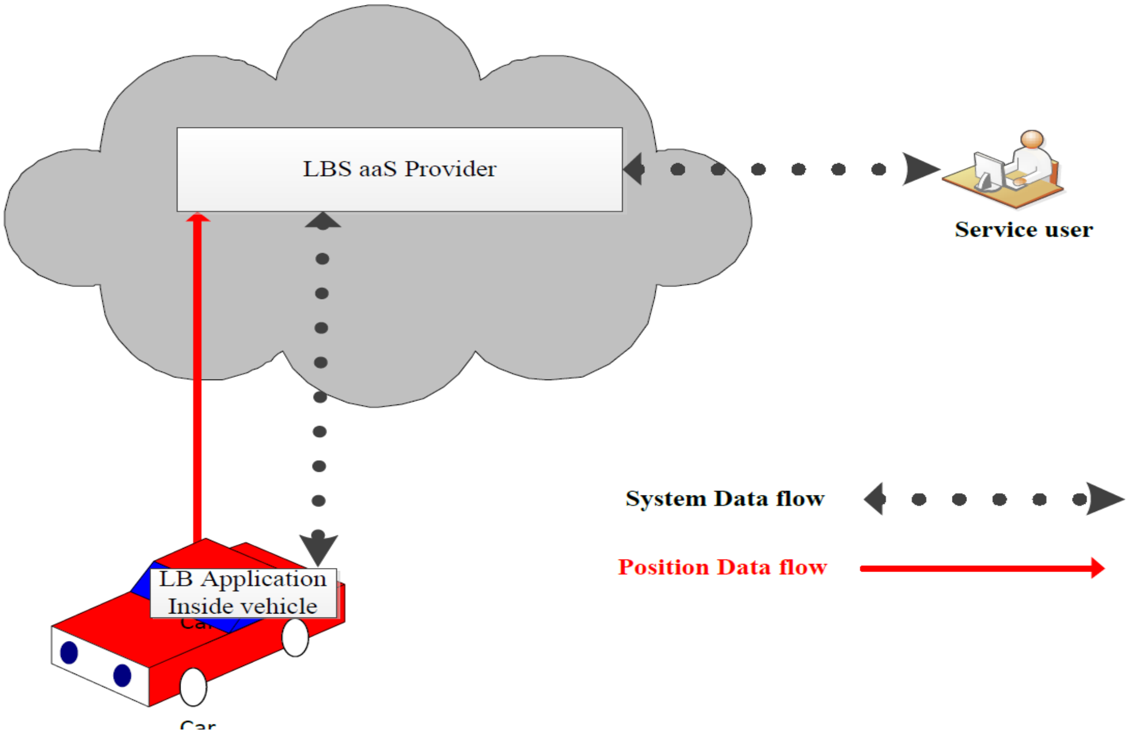

Figure 2-5 Location Based Service in VCC Data Flow

It's observed that, each LBS in VCC must have application client installed in vehicle operation system and application server in cloud application layer. As shown in figure 2-5, in order to produce LBS (Location-Based Services), the application server in cloud application layer must get accurate and real-time vehicle position data. The application client does three main functions:

1- It is send location update message to location server periodically.

2- It is waiting to receive instructions from application server.

3- It is executing the instructions as soon as receive them.

It can be observed that sending location update message to location server periodically consumes high bandwidth and cause overhead on server.

2.4 Summary

This thesis presents TSVCC, an optimized architecture, platform as a service that supports real-time vehicle applications that are “always-on” and in continuous communication with a server, as in traditional IP-based networks. TSVCC focuses primarily on impact the increasing number of location-based applications in same single vehicle but also discusses the structure of communication with the tracking unit and TSaaS components that support the overall framework. Unlike the other known architectures discussed in this chapter, TSVCC meets the needs of intelligent real-time vehicle applications in location-based services as discussed in this chapter. Our research presents the results of field tests in Chapter 4 which evaluate key TSVCC modules in order to quantitatively assess their impact on vehicle bandwidth utilization, and number of location update messages should be sent to server(storage capacity needs) in the context of the presented architecture.

Chapter 3

Proposed Tracking Service in Vehicular Cloud Computing (TSVCC) Platform as a Service

- The paper based on Chapter 3 has been published in the Proc. of 1st Scientific Conference on Information Technology SCITN, Sana'a, Yemen, 1-2 Nov 2016, Pp. 110-123.

CHAPTER 3 PROPOSED TRACKING SERVICE IN VEHICULAR CLOUD COMPUTING (TSVCC) Platform as a service

With the help of the VCC (Vehicular Cloud Computing), tens of billions of computing resources are just a mouse click away. These resources must all be tracked, but unsurprisingly, cannot all be tracked because vehicles are changing their locations on the road rapidly. However, there is a possibility to track all the vehicular cloud computing resources but still be having much network bandwidth consume. The next sections present the proposed Tracking Service in Vehicular Cloud Computing (TSVCC) platform as a service and it describe components of the proposed system.

3.1 Location Based Service in VCC

Location-Based-Application (LBA) /Location-Based-Service (LBS) is a type of vehicular cloud computing services and applications that depends on real-time vehicle's location. LBA systems should tracking vehicles' location and maintaining an accurate up-to-date view of the entire vehicular network. While location-based system is operating in VCC environment. It should cause data traffic flow between its components. The main goal is to split the data traffic to two main types: *Position update data flow* and *System data flow*.

- ***Position Update Data Flow*** is only data contain location information, each vehicle get own position data by GPS (Global Position System) model or any related technique discussed in chapter 1.
- ***System Data Flow*** is all non-location information that is required for the successful operation of the application (e.g., video, image, application logic parameters), also it is a data produce to service users.

In our proposed approach in this thesis, the position update data flow is only between tracking unit inside vehicle layer and TSaaS unit inside cloud platform layer to eliminate redundancy in location update messages in order to enhance performance of location-based applications.

As shown in figure 3-1, a single vehicle has three LBACs (Location-Based Application Client), each LBAC sends its own LUM (Location Update Message) which are have same content to server, and in this case it raised redundancy of sending LUM.

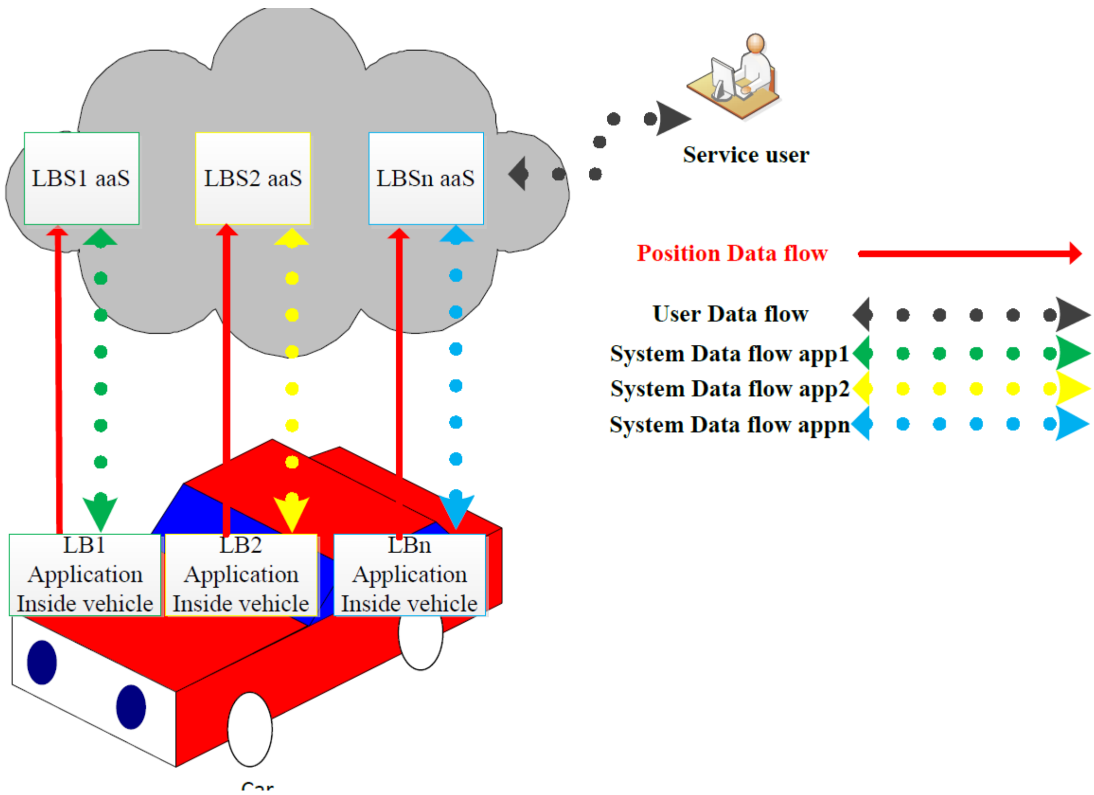

Figure 3-1 Single Vehicle Has Three LBACs

This thesis proposes solution to this problem and open future work in vehicular cloud computing area, especially location-based services.

3.2 Overview of Proposed Optimized VCC Architecture

This section presents overview of the proposed optimized VCC Architecture to which we call as Tracking Service in Vehicular Cloud Computing (TSVCC). TSVCC was created to meet application needs for real-time, high-accuracy and high precision location-based applications. This architecture was designed to be fully implementable by third party vehicle app developers, and can intelligently manage limited network communication layer. TSVCC can support various types of location-based applications, including real-time tracking, as well as delay-tolerant applications that record the user's travel path. For maximum flexibility, an application can dynamically manipulate TSVCC module parameters according to real-time application needs, and therefore hybrid applications with both real-time and delay-tolerant features are also possible.

To support the needs of modern LBS discussed in Chapter 2, TSVCC is separated into tracking unit modules, which are implemented in software on the vehicle OBU, in inside-vehicle layer and TSaaS modules, which reside on cloud platform layer.

Figure 3-2 shows the high-level view of this TSVCC architecture. The main goal of this section is to present an overview of proposed optimization vehicular cloud computing architecture. In order to improve the performance of location-based services and decrease network bandwidth consume while service users use services through vehicular cloud computing.

The number of location update messages sent from vehicles should be reduced to minimum. This chapter presents TSVCC to prevent impact of increasing number of location-based applications on same single vehicle.

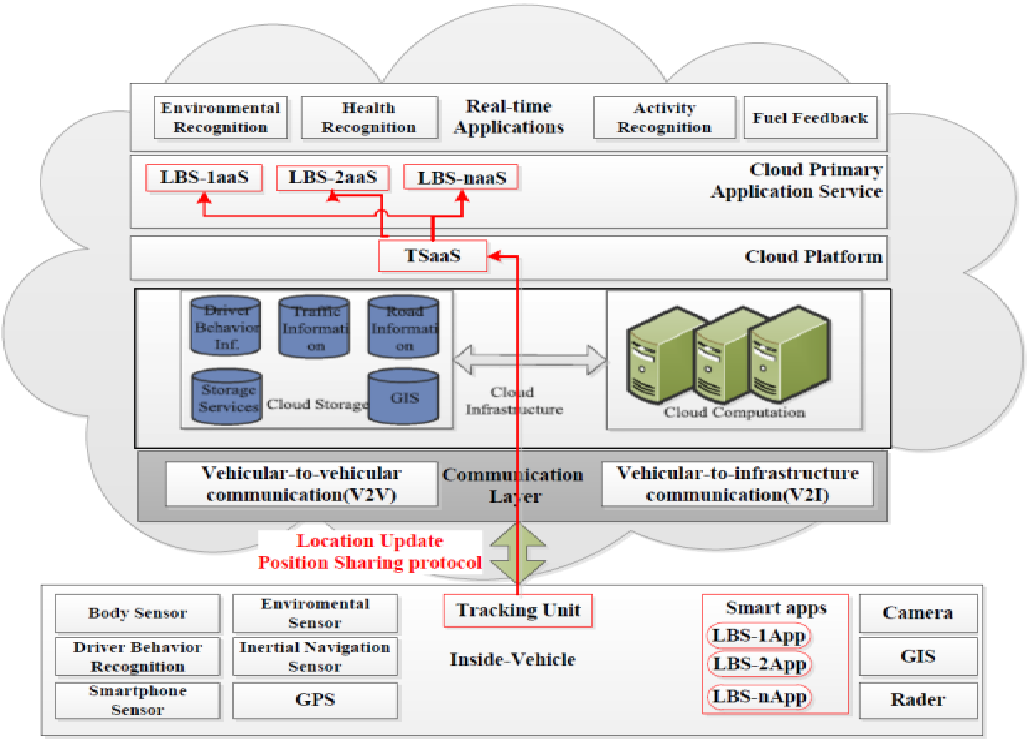

Figure 3-2 Optimized VCC Architecture

Now, the two major proposed components that overview the proposed architecture of vehicular cloud computing network, as illustrated in Figure 3-2, are namely:

- ***Tracking Unit***: this is software component found in inside-vehicle layer, it is responsible on calculate real-time position data and construct LUM (Location Update Message) then sending LUM to TSaaS unit in platform layer as shown in figure 3-2.
- ***TSaaS Unit*****:** this component found in cloud platform layer as shown in figure 3-2, it has three main subcomponents UDP server, database server and web server, which will be detail discussing in next section. It is responsible on receive location update messages, then it save them in location database server, also it use web server as interface to consume the proposed platform as a service. The set of web services are imple-

mented which can be used on demand by location-based applications developers.

Two-tier protocols, using both application-layer and transport-layer protocols in the same application, have been utilized in the past to increase VoIP performance for mobile devices [26], but have not been used in previously presented location-based service architectures. Splitting application logic and location data has architectural advantages in addition to bandwidth consume advantages, such as allowing easier integration of location data. We will explain our approach in detailed in following sections of this chapter.

3.3 Proposed System Architecture Components

In this section, the components that describe the proposed system architecture, as illustrated in Figure 3-3 are explained.

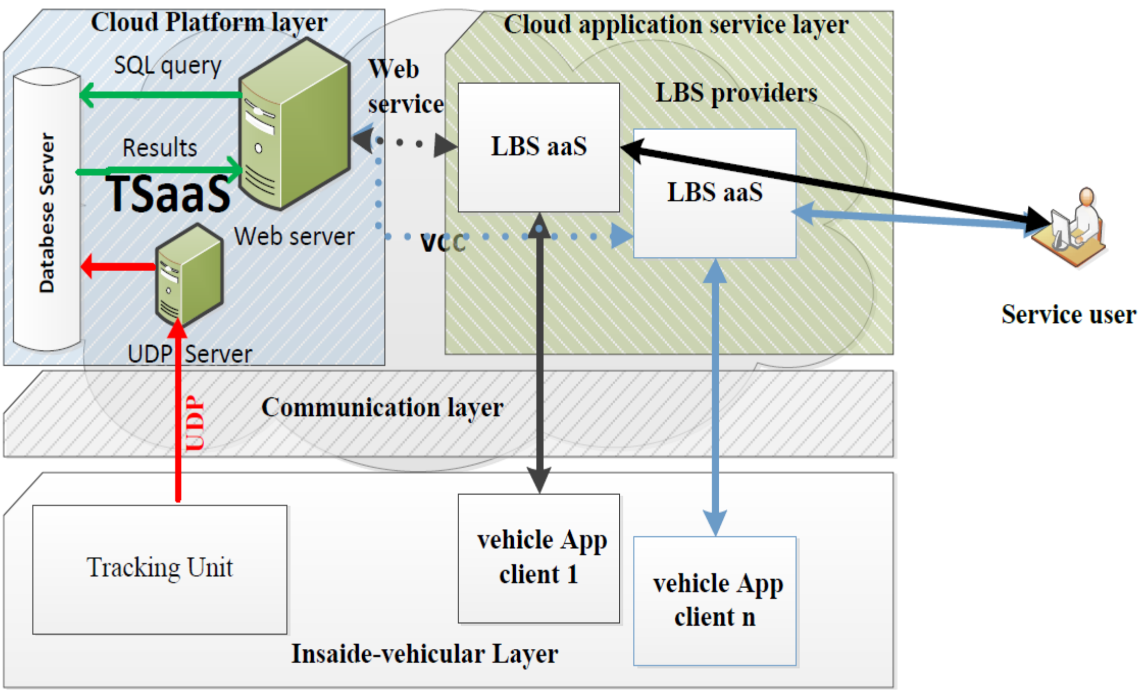

Figure 3-3 Proposed System Architecture Components

3.3.1 Tracking Unit

Tracking Unit is the software component that runs on the OBU (On-Board Unit) of the vehicle. It computes the current position of the vehicle (GPS Coordination) [8]. Then, periodically, Tracking Unit sends the position data to the UDP Server of TSaaS Unit with extra information such as vehicle identifier number, status, speed, and temperature. The main goal from tracking unit is get real-time vehicle's location . As shown in Figure 3-4, the Tracking Unit has two main functions:

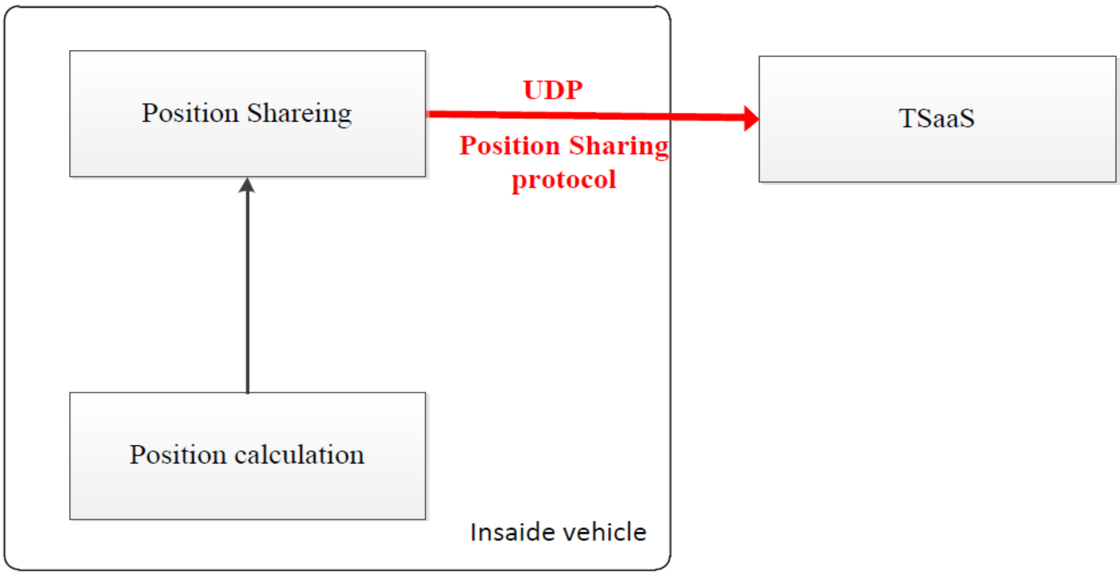

Figure 3-4 Tracking Unit System Model

Position Calculation**:** it is one of the basic function in the modern vehicle, due to the availably of many positioning techniques explained previously such as GPS. In our proposed research we used GPS because the availability of a satellite based global position system (GPS) and enhances the accuracy of position of vehicle in high dynamic environment (PMB-648 GPS receiver accuracy is 2 meters) [8].The vehicle identification number (VIN) is used as a unique Id for each a vehicle. The VIN is used for distinguished each a vehicle from other in our system. The tracking unit should be installed in the Vehicle. It can be implemented as a part of the vehicle operating system. The Tracking Unit computes the current position using GPS. Then, it adds the VIN and any information to position data and when location update massage is ready, the Tracking Unit start the next function which we refer as position sharing.

- ***Position Sharing*** is operation of sending the location update message from inside-vehicle layer to UDP server of TSaaS unit in cloud platform layer using UDP protocol [28] or any position sharing protocol [42]. The UDP protocol is used because it is connectionless and need less time to send message to server. Tracking unit has the IP address and port numbers of the UDP server in TSaaS unit at cloud platform layer. The tracking unit can support more than one UDP server through multicast address. We chose UDP, which is typically used for services where timeliness is favored over reliability (e.g., VoIP), as the protocol for efficient real-time location data transfer for TSVCC. The TSVCC platform treats streaming location data similarly to multimedia data in order to efficiently deliver timely location data from one entity in a location-based application system to another. The choice of UDP for location data transport differs from previous location-based architectures, largely because TSVCC is designed to meet the needs of real-time location-based applications that are always on.

3.3.1.1 Real-Time Vehicle’s Location Update

Real-time vehicle’s location update means that every vehicle updates its location in the real-time. Tracking unit is doing tow functions together (position calculation and position sharing) continually. Based on the concept shown in Figure

3-4, the real-time vehicle's location algorithm is implemented as shown in Algorithm 1 of the real-time vehicle's location algorithm.

The algorithm starts when a vehicle turn on then algorithm start initialization (a functions like connect with TSaaS unit in cloud platform layer, check software update and get performance control parameters), after that algorithm is computes VIN(Vehicle Identifier Number), it is assumed that every vehicle has unique serial number can read it by software, after that algorithm start interior loop while algorithm dose not interrupt then algorithm calculate real-time location after that algorithm is construct location update message then send it to TSaaS unit in platform layer via UDP protocol.

Algorithm 1:Real-time Vehicle's location

Begin

While (tracking unit switch on)

Initialization// check software update, get parametrats

Var VIN=Get Vehicle serial number

While (true)

Var POS=Get Current location // by GPS

Send current location to servers (VIN, POS) // by udp protocol

End while

End while

End

3.3.2 LBSaaS(Location-Based Service as a Service)

LBSaaS(Location-Based Service as a Service) is an application service in cloud application layer. It is consider on a kind of software's as a service leverage on real-time vehicle's location in VCC environment such Pic-on-wheel[11], Traffic analysis[12], Tracking systems[8], and Witness as a service[6], these services produce to the service users by LBS providers. It considers on web site or applet or application program.

3.3.3 TSaaS(Tracking Service as a Service)

TSaaS(Tracking Service as a Service) considers on a platform inside the cloud platform layer in VCC (Vehicular Cloud Computing). Based on the concept shown in Figure 3-3, TSaaS has three main components: Web server, Database server, UDP server, the follow subsection describe the subcomponents of TSaaS platform.

3.3.3.1 Database Server

Database Server considers on software component with storage capacity which used for storing the vehicle's location with extra information in order to be easy to store large information and retrieve it easily. UDP server is the only one component can write on Database server. Web server can only retrieve information from Database server. We can use oracle database server, MS SQL server, MySQL, etc.

3.3.3.2 Tracking Server (UDP server)

Tracking Server (UDP server) considers on software component that just have one main function, it receives the location updated messages from vehicle's tracking unit via UDP protocol. If the location update message is valid then it save its information in the database server else it drop the message.

3.3.3.2.1 Location Update

Tracking server (UDP server) receives location update messages from tracking unit of vehicles by UDP protocol [28]. Each message contains VIN (Vehicle Identifier Number) and position data. Periodically, a vehicle sends location update message to tracking server (UDP server) of TSaaS.

After receiving a LUM (Location Update Message) UDP server checks a validity of the received message using following DTD schema

```
<!—Location Update Message DTD file-- >
<!ELEMENT Location_Message(timestamp,Vehicle_Info+,Location_Info+,
Velocity)>
<!ELEMENT UTC_DateTime (#PCDATA)>
<!ELEMENT Vehicle_Info (VIN)>
<!ELEMENT VIN (#PCDATA)>
<!ELEMENT Location_Info (Lati,Long)>
<!ELEMENT Latitude(#PCDATA)>
<!ATTLIST LatitudeIndicator (S | N ) >
<!ELEMENT Longitude (#PCDATA)>
<!ATTLIST LongitudeIndicator (E | W ) >
<!ELEMENT Velocity (#PCDATA)>
```

Figure 3-5 Location Update Message DTD Schema

DTD schema used by Tracking Server(UDP server) for validating the location update message [9], when location update message has been arrived from tracking unit, tracking server(UDP server) checks a validity of the LUM by DTD schema that shown in Figure 3-5.

If LUM have not error and is valid, then tracking server (UDP server) saves elements of the LUM to the database server. The follow algorithm 2 describes the location update steps.

In algorithm 2, the algorithm runs through the following steps:

1. Initially, algorithm is waiting for receive LUM
2. When algorithm receive LUM save it in variable In *InComeLUM*.
3. if *InComeLUM* is not valid then go to step 1
4. extract elements of LUM , and store them in Database server

5. go to step 2

<u>Algorithm 2 location update</u>

Begin

While (true)

Var InComeLUM =Receive Location update message

IF InComeLUM is valid=true then

Extract message elements

Store the elements in new record in database server

End if

End while

End

3.3.3.3 Web Server

Web server is considered on software component that it just used to provide a set of web services as part of cloud platform layer. These set of web services is considered on location-based libraries to be used and consumed by platform as a service model. We implemented three web services as example to reader in chapter 4. It shows how we can be use of web service libraries. In future work can be add more libraries to our platform. This web service is consume by third-party location-based providers which it mentioned in above section LBSaaS. The web application server supports a large number of clients simultaneously and tracks individual sessions for each client.

3.3.3.3.1 Location Server Sharing (Platform Tracking System)

The proposed system uses a web server to handle the web service requests from LBSaaS providers. There are many web services can be provided to be used by applications developers to develop applications that depend on real-time vehicle's location over vehicular cloud computing. All web services provided by TSaaS dose common operations which convert a request of the LBSaaS unit to a suitable SQL query and execute it in database server then convert the results to suitable response. It implemented in chapter 4. The following algorithm 3 show web service request handling.

Algorithm 3 Tracking Web Service

Begin

Var request =Receive LBS provider request

Convert LBS provider request to suitable SQL query

Run SQL query on database

Convert result to LBS provider response

Send Response to LBS provider

End

In algorithm 3, the algorithm runs through the following steps:

1. Initially, algorithm is receive *LBS provider request*
2. When algorithm receive *LBS provider request* convert it into suitable SQL query
3. execute SQL query on database .and construct suitable response
4. Send Response to LBS provider
5. end

3.3.4 Vehicle Application Client

Vehicle application client is software installed inside a vehicle. For each LBSaaS application server has vehicle application client. The main function of vehicle application client is execute the instructions that is come from application server such as do images capture, video capture, or any other request then it send response to application server. The single vehicle can have many LBAC (Location-Based Application Clients). In the previous system, the application client is also periodically sends location update message to its own server location. In proposed system we remove this function from app-client's responsibilities and assign it to proposed tracking unit. Then, we proposed one location server for all applications server. This share the location server with all application servers that works on the same platform using service oriented architecture.

3.3.5 Service User

Service user is the end consumer, it is consumed utility of LBSaaS provider services on demand model, this user may be inside vehicle or anywhere connected to internet. It’s any one interest with obtain on service related on vehicle’s location .

3.4 Conclusion

In this chapter, we propose an optimization to VCC (Vehicular Cloud Computing) architecture. The begin of this chapter is introduction about location-based service data flow and location update problems in vehicular cloud computing. New solution is proposed to support developing many LBS applications in VCC. Components of proposed system architecture discusses in details. The implementation and evaluation of proposed system is presented in chapter 4.

Chapter 4

Implementation and Evaluation

CHAPTER 4 IMPLEMENTATION AND EVALUATION

In previous chapter, Tracking Services in Vehicular network based on Cloud Computing (TSVCC) platform as a service is presented. The proposed system components are detailed. This chapter is divided into two parts:

1. Implementation of individual TSVCC components.
2. Evaluation of the proposed system and results discusses.

The first section of this chapter presents implementation of TSVCC components in order to prove the proposed method. The second section evaluates the proposed system and discusses the results.

4.1 Implementation Of Individual TSVCC Components

This section will show implementations of four different components: Database server, UDP server, Web server and Tracking unit.

4.1.1 Database Server Implementation

Database server is software component. It is used for create rational database in order to store and retrieve position data which arrived from vehicles. Microsoft SQL Server 2008 is used in our proposed system. As shown in Figure 4-1, we have used simple database in order to be understood by the reader.

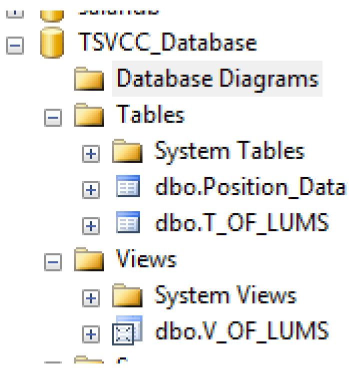

Figure 4-1 proposed database

The proposed database contains one table to save position data. We refer it T_OF_LUMS. As shown in figure 4-2, the T_OF_LUMS has set of columns which used to store elements of location update message.

T_OF_LUMS

Column Name	Data Type	Allow Nulls
TableIdent	bigint	☐
LUM_VIN	nvarchar(20)	☑
LUM_Latitude	float	☑
LUM_Longtude	float	☑
LUM_DT	datetime	☑
		☐

Figure 4-2 DFD of T_OF_LUMS table

The proposed database contains one view that called V_OF_LUMS. In order to retrieve position information from the proposed database, we use the view instead of the table because we want to prevent any update by web server.

4.1.2 UDP Server Implementation

UDP server is software component. It is used for handling all LUM messages which arrived from tracking unit. Microsoft Visual Studio 2010 is used to develop the proposed UDP server. UDP server is implemented by visual basic console application project. The ip address and port number should be entered manually when UDP server startup. In order to enable UDP server receives the location update messages. After receiving the location update messages, UDP server sends them elements to database server. The UDP server can receive many LUM messages at the same time using UDP protocol and then save their elements to database server. Figure 4-3 show UDP server while it is running.

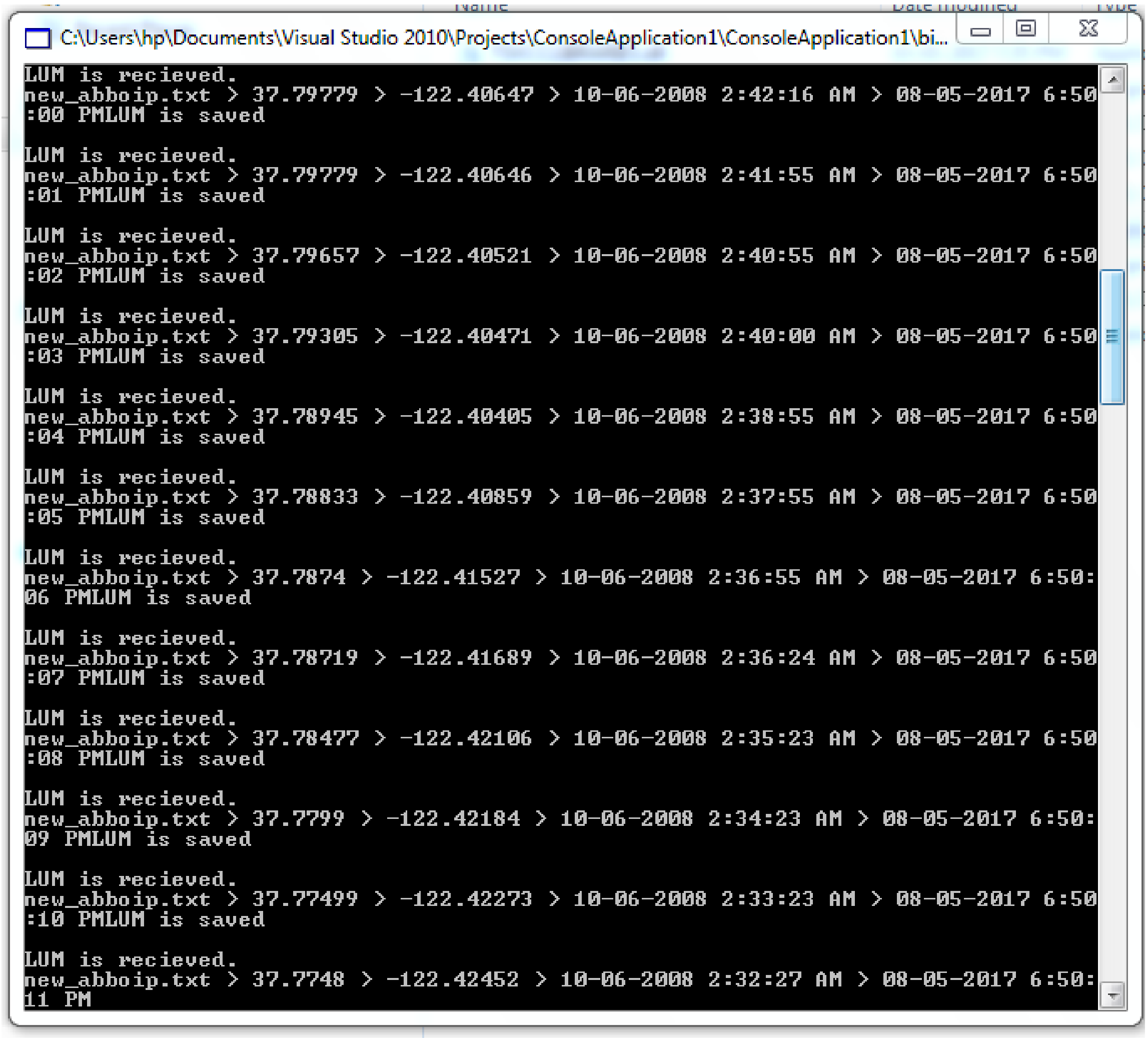

Figure 4-3 Screen Shot while UDP server running

4.1.3 Web Server Implementation

Web server is software component. It used for create web services interface of proposed system. Some web services implemented by ASP.NET 2010. The implemented web services can retrieve position information from the database server. We have implemented three web methods called: Vehicle_After_DateTime, Vehicle_All_LUMs and Vehicle_Last_LUM. We have implemented these methods according to needs of most location-based applications. More web methods can be added in future research works. It needs study and analysis LBA requirements.

4.1.3.1 Vehicle_After_DateTime

This method is implemented to consume by third-party location-based application developers, it is suitable for Pic-on-Wheels [11]. This method has two parameters (VIN=Vehicle Identifier Number and AF_DT= a specific date/time). It returns the list of location update messages which match with a given VIN and after a given AF_DT specific date/time. The following is a sample SOAP 1.2 request. The placeholders shown need to be replaced with actual values.

```
POST /Service1.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Vehicle_After_DateTime xmlns="http://tempuri.org/">
      <VIN>string</VIN>
      <AF_DT>dateTime</AF_DT>
    </Vehicle_After_DateTime>
  </soap12:Body>
</soap12:Envelope>
```

The following is a sample SOAP 1.2 response. The placeholders shown need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Vehicle_After_DateTimeResponse xmlns="http://tempuri.org/">
      <Vehicle_After_DateTimeResult>
        <LUM>
          <VIN>string</VIN>
          <LATITUDE>string</LATITUDE>
          <LONGITUED>string</LONGITUED>
          <SND_DT>dateTime</SND_DT>
          <RCV_DT>dateTime</RCV_DT>
        </LUM>
        <LUM>
          <VIN>string</VIN>
          <LATITUDE>string</LATITUDE>
          <LONGITUED>string</LONGITUED>
          <SND_DT>dateTime</SND_DT>
          <RCV_DT>dateTime</RCV_DT>
        </LUM>
      </Vehicle_After_DateTimeResult>
    </Vehicle_After_DateTimeResponse>
  </soap12:Body>
</soap12:Envelope>
```

4.1.3.2 Vehicle_All_LUMs

This method is implemented to consume by third-party location-based application developers, it is suitable for Traffic Analysis as a Service [6]. The parameter for this method is vehicle identifier number. It returns the list of location update messages which match with a given VIN. The following is a sample SOAP 1.2 request. The placeholders shown need to be replaced with actual values.

```
POST /Service1.asmx HTTP/1.1
```

```
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Vehicle_All_LUMs xmlns="http://tempuri.org/">
      <VIN>string</VIN>
    </Vehicle_All_LUMs>
  </soap12:Body>
</soap12:Envelope>
```

The following is a sample SOAP 1.2 response. The placeholders shown need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Vehicle_All_LUMsResponse xmlns="http://tempuri.org/">
      <Vehicle_All_LUMsResult>
        <LUM>
          <VIN>string</VIN>
          <LATITUDE>string</LATITUDE>
          <LONGITUED>string</LONGITUED>
          <SND_DT>dateTime</SND_DT>
          <RCV_DT>dateTime</RCV_DT>
        </LUM>
        <LUM>
```

```
          <VIN>string</VIN>
          <LATITUDE>string</LATITUDE>
          <LONGITUED>string</LONGITUED>
          <SND_DT>dateTime</SND_DT>
          <RCV_DT>dateTime</RCV_DT>
        </LUM>
      </Vehicle_All_LUMsResult>
    </Vehicle_All_LUMsResponse>
  </soap12:Body>
</soap12:Envelope>
```

4.1.3.3 Vehicle_Last_LUM

This method is implemented to consume by third-party location-based application developers, it is suitable for Tracking System [8]. The parameter for this method is vehicle identifier number. It returns the last of location update message which match with a given VIN. The following is a sample SOAP 1.2 request. The placeholders shown need to be replaced with actual values.

```
POST /Service1.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Vehicle_Last_LUM xmlns="http://tempuri.org/">
      <VIN>string</VIN>
    </Vehicle_Last_LUM>
  </soap12:Body>
</soap12:Envelope>
```

The following is a sample SOAP 1.2 response. The placeholders shown need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Vehicle_Last_LUMResponse xmlns="http://tempuri.org/">
      <Vehicle_Last_LUMResult>
        <VIN>string</VIN>
        <LATITUDE>string</LATITUDE>
        <LONGITUED>string</LONGITUED>
        <SND_DT>dateTime</SND_DT>
        <RCV_DT>dateTime</RCV_DT>
      </Vehicle_Last_LUMResult>
    </Vehicle_Last_LUMResponse>
  </soap12:Body>
</soap12:Envelope>
```

4.1.4 Tracking Unit Implementation

Tracking unit is software component. Periodically, it sends LUM messages to UDP server. Microsoft Visual Studio 2010 is used to develop the proposed tracking unit. Tracking unit is implemented by visual basic windows application project. The ip address and port number of UDP server should be entered manually when tracking unit startup in order to enable tracking unit sends the location update messages.

Tracking unit use a position data which available in cabspotting dataset [41]. This data set contains mobility traces of taxi cabs in San Francisco, USA. It contains

GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area. Cab mobility traces are provided by the Exploratorium the museum of science, art and human perception through the cabspotting project: http://cabspotting.org. Each taxi is equipped with a GPS receiver and sends a location update message (timestamp, identifier, geo-coordinates) to a central server. The average time interval between two consecutive location updates is less than 10 sec, allowing us to accurately interpolate node positions between location updates. Figure 4-4 shows tracking unit while it is reading position data from cabspotting dataset and sending location update messages to the UDP server.

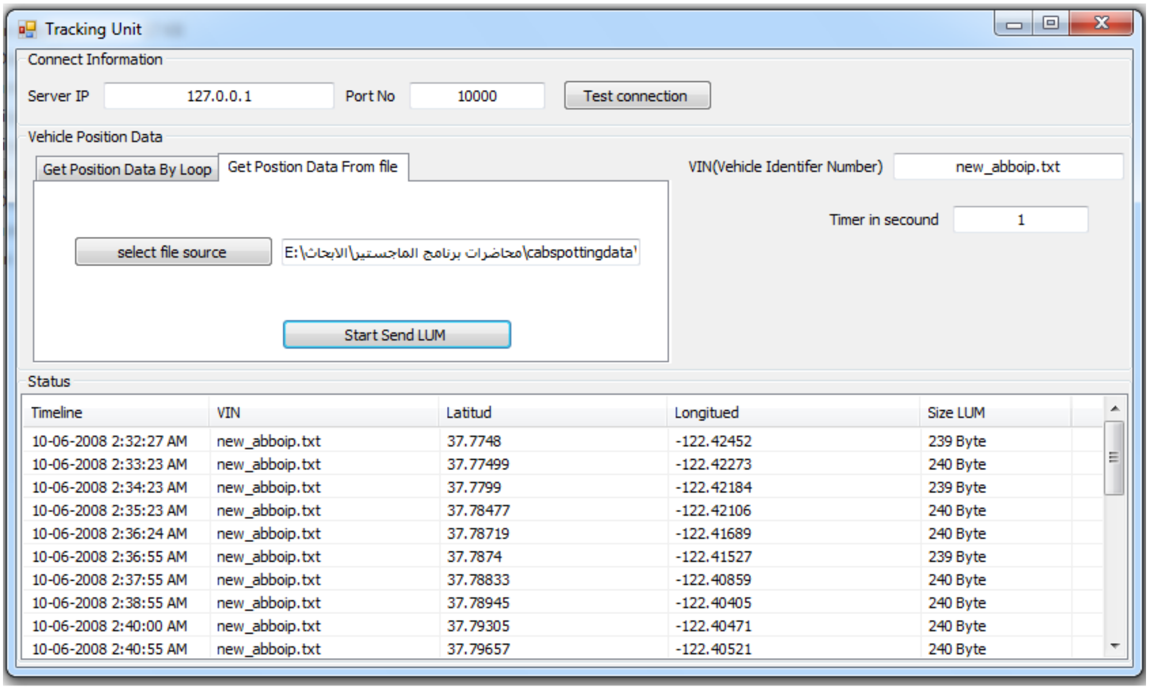

Figure 4-4 Screen Shoot while Tracking Unit running

4.2 Evaluation

The main goal from this section is to evaluate the proposed system that implemented in pervious section and discussing the results. The following subsections analyze the performance of the proposed optimized architecture and compare the performance of the proposed scheme with the performance of ordinary VCC architecture.

In this thesis, the proposed system is evaluated through studying the relationship between the number of LBAC(Location-Based Application Client) with number of location update messages should send to location server (storage capacity), also in following sections we investigate impact of increasing number of LBAC(location-based application client) within same single vehicle on communication overhead (bandwidth utilization) [10] .

It assumes a single vehicle subscribed to VCC capability to have multi task operation system like android mobile system, so any single vehicle subscribed to VCC can has more than one LBA.

4.3 Hypothesizing

1- Increasing number of LBSCs inside single vehicle lead to increasing of number of location update messages should flow cross communication layer in VCC which cause increasing of storage capacity.

2- Increasing number of LBSCs inside single vehicle lead to increasing bandwidth utilization.

4.4 Quantativly Evaluation

The process of identifying and obtaining the results was conventional. Let consider *Size_LUM* denote to the minimum necessary bytes for carry vehicle's location information such as (GPS data, VIN and speed), which referred Location Update Message (LUM). As shown in figure 4-4, the size of LUM is 240 Byte, so one UDP packet can carry one location update message [11]. It assumes a vehicle have wireless NIC (Network integrated circuit) 54 Mb/s like Laptop device.

Let *NLBAC* denote to the number of LBAC's installed on a same single vehicle which subscribed to VCC. Let *SC* denote to the storage capacity consumed by single vehicle while its LBAs update the location in real-time. Then, the total size of location update messages by single vehicle is:

*SC=NLBAC * Size_LUM   Byte…………………………………………………….(1)*

After that, it is easy to computes the total vehicle bandwidth utilization while LBAs update the location in the real-time.

*BU=NLBAC * Size_LUM /vehicle network card bandwidth……….(2)*

4.5 Results and Discussion

This section explains the performance of the proposed optimized architecture with comparing the performance of ordinary VCC architecture. After implementing the proposed system as shown at begin of this chapter, the proposed system is tested for five minutes. We again the testing with change number of LBSC and the results are collected. The results used as input to pervious equations. The results discussed in two metric: storage capacity, bandwidth utilization.

As shown in figure 4-5, it presents the impact of increasing number of LBACs in a single vehicle on number of necessary location update massages (storage capacity used on location servers). Location server has limited storage capacity which used to store location update messages. Note that, in previous system when number LBAC increase, the number of necessary location update massages increase but in proposed system, it stops the impact of increasing number of LBAC on a same single vehicle.

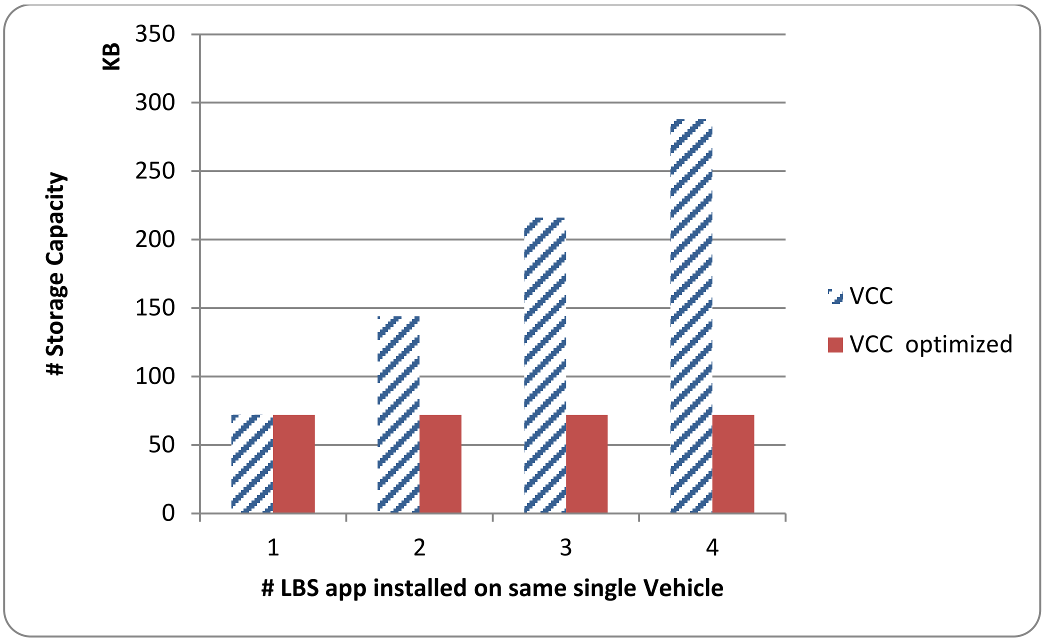

Figure 4-5app client vs. Storage Capacity

As shown figure 4-6, it presents the impact of increasing number of LBACs in single vehicle on network bandwidth utilization. Note that, in previous system when number of LBACs increase, the vehicle bandwidth utilization increase but in proposed system, it stops the impact of increasing number of LBACs on a same single vehicle.

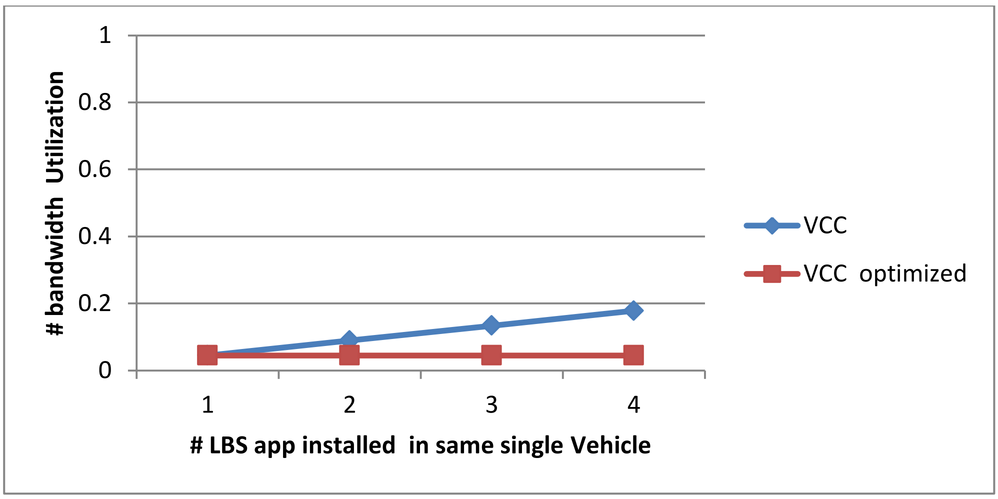

Figure 4-6 app client vs. bandwidth utilization

4.6 Conclusions

In this chapter, implementation a novel enhancement to VCC architecture is proposed. The proposed scheme supports development real-time vehicle's location applications in the intelligent transport system which produced by VCC model. The contributions of this thesis are that it to solve the problem of huge bandwidth consume required by previous model. It open future work in many issues such as add identification service to platform, optimize security and privacy, open area to researchers to add new functions for VCC platform and can create new LBS applications in future.

Chapter 5

Conclusions and Future Works

CHAPTER 5 CONCLUSIONS AND FUTURE WORKS

Location-Based Services(LBS) in vehicular cloud computing is interest, but location-based applications in VCC face with location update problem, many researches provided solution to location update problem which take in account each a single vehicle contain a single LBA but this not enough in real-life. We can observe that, in the current system of smart phone, a single phone has many LBAs .VCC has been shifted from Mobile Cloud Computing (MCC), so in real life a single vehicle may be containing more than one of LBA on the same vehicle. So the traffic in vehicular networks overhead can be unmanageable when thousands of vehicles must update their locations within short time. It is proposed to develop efficient Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) platform as a service for supporting third-party LBAs in vehicular networks and handling impact of increasing number of LBAs on a single vehicle. It should have the following features:

- Less overhead bandwidth.
- Less number of location update massages(less storage capacity need).
- Efficient support third-party LBAs.

This thesis presents a novel Tracking Service in Vehicular networks based on Cloud Computing (TSVCC) platform as a service, full implemented. It presents an optimized VCC architecture. The proposed system architecture meets the needs of intelligent real-time vehicle applications and is fully implementable by third-party vehicular application developers.

In this thesis, a novel enhancement to the VCC architecture is proposed. The proposed scheme supports development real-time vehicle’s location applications which deploy by VCC model in the intelligent transport system. The contributions of this thesis are that it is to solve the problem of huge bandwidth consume required by previous model. It open future works in many issues such as add identification service to platform, optimize security and privacy, open area to researchers to add new functions for VCC platform and can create new LBS applications in future.

References

[1] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, Kun Yang, “Toward Cloud-based Vehicular Networks with Efficient Resource Management”, *IEEE Network*, (Volume:27 ,Issue: 5), Oct 2013, pp. 48 - 55.

[2] Fekri M.Abduljalil, "Video Capture Service in the Intelligent Transportation System based on Cloud Computing", *International journal of Computer Applications(0975-8887)*,Volume 97-No.5,July 2014.

[3] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, Rajkumar Buyya, "A survey on vehicular cloud computing",*Journal of Network and Computer Applications(2013)*, http://dx.doi.org/10.1016/j.jnca2013.08.004

[4] Mario Gerla," Vehicular Cloud Computing", *Vehicular Communications and Applications Workshop* ,2012 IEEE

[5] P Pesti, L Liu, B Bamba, A Iyengar, M Weber “ROADTRACK: Scaling Location Updates for Mobile Clients on Road Networks with Query Awareness”, *Proceeding of the VLDB Endowment*, Volume 3, Issue 1-2,September 2010.

[6] Rasheed Hussain, Fizza Abbas, Junggab Son, Donghyun Kim, Sangjin Kim, Heekuck Oh, “Vehicle Witnesses as a Service: Leveraging Vehicles as Witnesses on the Road in VANET Clouds”, *in Cloud Computing Technology and Science (CloudCom)*, 2013 5th International Conference on, vol. 1, pp. 439-444. IEEE, 2013.

[7] Won Min Kang , Jae Dong Lee , Young-SikJeong and Jong Hyuk-Park,"VCC-SSF: Service-Oriented Security Framework for Vehicular Cloud Computing", *Sustainability 2015*, 7, 2028-2044; doi:10.3390/su7022028

[8] Nagarjuna Karyemsetty, Badugu Samatha, K Himantha Rao, " Design and Deployment of Vehicle Tracking System in VANETs using Xbee Pro: Prototype Model", *International Conference on Communication Networks (ICCN)*,2015

[9] J. G. Silveira, R. Balbinot, J. A. Oliveira Neto, F. C. Castello, A. R. Vieira, A. Quadra," Telemetry Extensions for the Mobile Location Protocol",*IEEE*,2003

[10] Sulata Mitra, "Dynamic Resource Reservation for Authentic Vehicles in VANET", *International Conference on Innovations in Engineering and Technology (ICIET'2013)* Dec. 25-26, 2013 Bangkok (Thailand)

[11] Mario Gerla, Jui-Ting Weng, Giovanni Pau,"Pics-On-Wheels:Photo Surveillance in the Vehicular Cloud" ,*IEEE International Conference on Computing*, 2013,pp.1123-1127.

[12] Victor W. Chu∗ , Raymond K. Wong∗ , Wei Liu†, Fang Chen†, Chang-Shing Perng," Traffic Analysis as a Service via a Unified Model", *2014 IEEE International Conference on Services Computing*.

[13] Abdessamed Derder, Samira Moussaoui, "Target Tracking in VANETs Using V2I and V2V Communication" ,*International Conference on Advanced Networking Distributed Systems and Applications*,2014

[14] S.Boopathi, K.Govindaraju, M.Sangeetha, M.Jagadeeshraja, M.Dhanasu, "REAL TIME BASED SMART VEHICLE MONITORING AND ALERT USING GSM", *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 3, Issue 11, November 2014

[15] Jetendra Joshi, Kritika Jain and YashAgarwal," CVMS: Cloud Based Vehicle Monitoring System in VANETs ", *International Conference on Connected Vehicles and Expo (ICCVE)*,2015.

[16] Research and Innovative technology administration (RITA), (USDOT) U S Department of transportation 1200 New Jersey Avenue.

[17] ERTICO Intelligent Transport Systems and Services stakeholders in Europe: http://www.ertico.com/ertico-its Europe, last accessed on June 10th, 2016.

[18] Ministry of Land, Infrastructure, Transport and Tourism in Japan: http://www.mlit.go.jp/en/index.html, last accessed on June 10th, 2016.

[19] Intelligent Transportation Systems Society of Canada: http://www.itscanada.ca, last accessed on June 10th, 2016.

[20] Zhao Yilin (2002), "Standardization of mobile phone positioning for 3G systems," *Communications Magazine*, IEEE, Vol. 40 pp. 108-116.

[21] GPS World. (2008). "GPS handset market poised for huge expansion" GPS World. Accessed: May 9, 2016 from http://www.gpsworld.com/wireless/news/abi-gps-handset-market-poised-hugeexpansion- 3864

[22] D. Porcino (2001), "Location of third generation mobile devices: a comparison between terrestrial and satellite positioning systems," *in Vehicular Technology Conference*, 2001. VTC 2001 Spring. IEEE VTS 53rd, pp. 2970-2974 Vol. 4, 2001.

[23] Sean J. Barbeau," A Location-Aware Architecture Supporting Intelligent Real-Time Mobile Applications", Doctor of Philosophy Department of Computer

Science and Engineering thesis work, College of Engineering, University of South Florida, June, 2012

[24] Skyhook Wireless, “Hybrid Positioning System (XPS),” http://www.skyhookwireless.com/howitworks/.

[25] A. Civilis, C. S. Jensen, and S. Pakalnis, “Techniques for efficient road-network-based tracking of moving objects,” *Proc. IEEE TKDE*, vol. 17, no. 5, pp. 698–712, 2005.

[26] J. W. Jung, R. Mudumbai, D. Montgomery, and Kahng Hyun-Kook (2003), "Performance evaluation of two layered mobility management using mobile IP and session initiation protocol," *in Global Telecommunications Conference,* 2003. GLOBECOM '03. *IEEE*, pp. 1190-1194 Vol. 3, 1-5 Dec. 2003.

[27] Internet Engineering Task Force, "Request for Comments (RFC) 2616 – Hypertext transfer Protocol -- HTTP/1.1," ed, 1999.

[28] Internet Engineering Task Force, "Request for Comments (RFC) 768 – User Datagram Protocol," ed, 1980.

[29] Internet Engineering Task Force, "Request for Comments (RFC) 793 – Transmission Control Protocol - DARPA Internet Program Protocol Specification," ed, 1981.

[30] Sean Barbeau, Rafael Perez, Miguel Labrador, Alfredo Perez, Philip Winters, and Nevine Georggi (2011), "A Location-Aware Framework for Intelligent Real-Time Mobile Applications," *Pervasive Computing*, *IEEE*, Vol. 10 pp. 58-67.

[31] Sean J. Barbeau, Nevine L. Georggi, and Philip L. Winters (2010), "Global Positioning System Integrated with Personalized Real-Time Transit Information from Automatic Vehicle Location," *Transportation Research Record: Journal of the Transportation Research Board*, pp. 168-176.

[32] Google, Inc. "My Tracks." Accessed December 16, 2016 from http://mytracks.appspot.com/

[33] Google, Inc. "Google Maps." Accessed December 16, 2016 from http://maps.google.com/

[34] Google, Inc. "Google Maps Navigation (Beta)." Accessed December 16, 2016 from http://www.google.com/mobile/navigation/

[35] Telenav. "Telenav GPS Navigator." Accessed December 16, 2016 from http://www.telenav.com/products/tn/

[36] INRIX. "INRIX - Go Anywhere." Accessed December 16, 2016 from http://www.inrix.com/

[37] Foursquare Labs, Inc. "foursquare." Accessed December 21, 2016 from http://foursquare.com/

[38] Facebook, Inc. "Facebook." Accessed December 21, 2016 from http://www.facebook.com/

[39] Google, Inc. "Latitude." Accessed December 21, 2016 from www.google.com/latitude

[40] Chen, H., Chiang, Y. Chang, F., H. Wang, H. (2010). Toward Real- Time Precise Point Positioning: Differential GPS Based on IGS Ultra Rapid Product, *SICE Annual Conference*, The Grand Hotel, Taipei, Taiwan August 18-21.

[41] Michal Piorkowski, Natasa Sarafijanovic Djukic, Matthias Grossglauser. CRAWDAD dataset epfl_mobility (v.2009 02 24). Feb 2009. http://crawdad.org/epfl/mobility/20090224 (accessed JUN 2017).

[42] Z. Riaz, "Optimized position update protocols for secure and efficient position sharing," Master's thesis, 2013.

[43] D. Bernstein, N. Vidovic and S. Modi, "A Cloud PAAS for High Scale, Function, and Velocity Mobile Applications - With Reference Application as the Fully Connected Car, *in Proc. Fifth International Conference on Systems and Networks Communications (ICSNC)*, pp. 117-123, 2010.

ملخص

خدمة التتبع في شبكات المركبات بالاعتماد علي الحوسبة السحابية

رسالة مقدمة لاستكمال متطلبات الحصول علي درجة الماجستير في **تقنية المعلومات**

جامعة الاندلس للعلوم والتقنية

اسم الطالب

صلاح محمد عبد الجليل علي

2017

زيادة اعداد الخدمات المقترحة في شبكات المركبات ساهم في بناء انظمة ذكية كبيره مثل المدن الذكية وانظمة النقل الذكية. الهدف الاساسي كان مساعدة الناس علي الطريق.

حديثا, العديد من الانظمة التي تعتمد علي الموقع اقترحت في الحوسبة السحابية للمركبات كتطبيقات تقوم بوظائف خاصه مثل انظمة حماية السيارات ضد السرقة وانظمة مراقبه المركبات علي الطرقات وادارة اساطيل السيارات وغيرها. هذا النوع من التطبيقات يعتمد علي مواقع المركبات في الزمن والمكان الحقيقي. ندعوها بالتطبيقات التي تعتمد علي الموقع او بالخدمات التي تعتمد علي الموقع.

هذه الرسالة تعرض خدمة تتبع في شبكات المركبات بالاعتماد علي الحوسبة السحابية "بيئة تطوير كخدمة". هذه الخدمة المقترحة تدعم تطوير عدة تطبيقات تعتمد علي الموقع في بيئة الحوسبة السحابية للمركبات. نحن اقترحنا تحسين لمعمارية الحوسبة السحابية للمركبات من اجل تحسين استخدام سعة الشبكة وتقليل عدد رسائل تحديث الموقع التي يجب ان تقوم المركبة بإرسالها الي سيرفر الموقع (السعة التخزينية في سيرفر الموقع). الطريقة المقترحة عالجت تأثير زيادة عدد التطبيقات التي تعتمد علي الموقع داخل المركبة,تم التقييم علي اساس نسبة استهلاك سعة الشبكة والسعة التخزينية للبيانات في سيرفر الموقع الي عدد التطبيقات التي تعتمد علي الموقع داخل المركبة. النظام المقترح تم تنفيذه باستخدام تقنيات الدوت نت و تم تقيم النظام المقترح ومناقشة النتائج

جامعة الاندلس للعلوم والتقنية
كلية الهندسة
قسم تقنية المعلومات

خدمة التتبع في شبكات المركبات بالاعتماد علي الحوسبة السحابية

رسالة مقدمة لاستكمال متطلبات الحصول علي درجة الماجستير في **تقنية المعلومات**

جامعة الاندلس للعلوم والتقنية
اسم الطالب
صلاح محمد عبد الجليل علي
اسم المشرف
فكري الحميدي, استاذ مشارك ,جامعة الاندلس

2017