國立交通大學

資訊科學與工程研究所

碩 士 論 文

一個基於車載隨意行動網路之行車導航系統的分析

The Analysis of Vehicle Navigation System Based on Vehicular Ad-Hoc Networks

研 究 生：吳忻霖

指導教授：張明峰　教授

中 華 民 國　102　年　7　月

一個基於車載隨意行動網路之行車導航系統的分析

The Analysis of Vehicle Navigation System Based on Vehicular Ad-Hoc Networks

研 究 生：吳忻霖　　Student：Hsin-Lin Wu

指導教授：張明峰　　Advisor：Prof. Ming-Feng Chang

國 立 交 通 大 學
資 訊 科 學 與 工 程 研 究 所
碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2013

Hsinchu, Taiwan, Republic of China

中華民國 102 年 7 月

一個基於車載隨意行動網路之行車導航系統的分析

研究生:吳忻霖　　　　　　　指導教授:張明峰教授

國立交通大學資訊科學與工程研究所 碩士班

摘要

提供駕駛者即時交通資訊各路段旅行時間，並配合導航系統的使用，能有效的節省行車時間及能源消耗。由於具備 GPS 定位系統及無線資料傳輸的智慧型手機日漸普及，利用行駛車輛上的智慧型手機收集並與其他車輛交換即時交通資訊，再以此資訊週期性的評估出目前的交通狀態，即時的選出最佳旅行路線成為可行的方式。在傳統的分散式系統，每台使用導航系統的車輛使用泛洪法週期性的廣播資料，這將會使網路中充斥許多不必要的交通採集樣本，造成頻寬的浪費。為了增加資訊傳輸效率，許多的方法因此而被提出。其中，流言法在高密度的流量下具有極佳的效能。因此一個使用優先權流言法的導航系統 CATE，擁有極佳的資訊傳播效率，並且透過一極為擬真的實驗證明了其效能和高度擴展性，但是這套導航系統仍有其不足之處，未考慮在十字路口所可能產生的延遲，以及可能會產生二度擁塞的問題。我們提出一個方法評估各路段的轉彎延遲來解決十字路口的延遲問題，透過 Veins 的擬真模擬結果顯示我們的方法可以額外再度減少 17%的平均旅行時間，並減少 2%因為使用導航系統反而造成旅行時間增長的車輛數。

The Analysis of Vehicle Navigation System Based on Vehicular Ad-Hoc Networks

Student: Hsin-Lin Wu Advisor: Prof. Ming-Feng Chang

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

Providing real-time traffic information, such as the travel time of each road segment, to road users and coupling this information with navigation system can save traveling time and reduce fuel consumption. With the increasing popularity of smartphones that are embedded with GPS receiver and wireless data communication capability, smartphones on journeying vehicles can collect and exchange the real-time traffic information, and then dynamically choose a time-saving route based on the collected traffic condition. In a conventional distributed system, navigation-embedded vehicles use the flooding method propagating the traffic information periodically. However, there may be redundant samples spreading in the network. For the purpose of improving the dissemination efficiency, many methods have been purposed. Gossip-based propagation models have a great efficiency in dense traffic. So the navigation system CATE, using utility-based gossip model to keep the scalability and the efficiency, has been proved to offer scalability in a realistic experiment. However, CATE has some deficiencies that need to be improved; for example, intersection delay is not considered in route planning and secondary congestion problem may occur after re-routing by users using the same navigation system. We purpose a method that provides each road segment’s turning delay to improve the travel time for more road users. Veins has been used to simulate our navigation system. The simulation results indicate our method can save more than 17% average travel time and reduce the percentage of cars that experience longer travel time by 2%.

誌謝

首先我要感謝指導教授 張明峰老師。老師在我的研究路上給了我很多的指引，從起初的構想、過程中遇到的困難、以及最後完整的架構，老師有條理、有耐心的指導讓我獲益良多,也使我最後得以完成碩士論文;更是藉由老師的指導,讓我學習到獨立思考以及對研究明確精準的態度,我相信這會是我一輩子受用的教導。再次謝謝老師兩年來的指導。

再來也要謝謝實驗室的同學們雅勤、嵐茵以及郁晴，還有上一屆的學長們，感謝你們在課業還有生活上的幫助，讓我順利完成研究所學業。也要感謝實驗室的學弟妹們的幫助，特別是建良和松霖幫忙一起做模擬、跑數據。感謝以上的陪伴與共同奮鬥，使我的研究所生活多采多姿。

最後，最感謝的是我摯愛的家人。感謝你們的支持，讓我順利完成學業。

吳忻霖 謹識於

國立交通大學資訊科學與工程研究所碩士班

中華民國壹零貳年七月

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Current Development

Since urban traffic has been growing all over the world in recent years, reducing travel time by using real-time traffic Information to avoid traffic congestion becomes an important issue for drivers. Traffic Information Systems (TIS) collect real-time traffic data from a variety of sensors, and integrate the data into accurate and reliable traffic information, and then distribute the information to road user for routing planning. TIS can improve transportation efficiency effectively. In addition, using an intelligent transportation system (ITS) that consists of traffic information system, car navigation, and traffic signal control system can further save much more travel time.

In order to collect traffic real-time data from the sensors, several methods have been developed. There are two categories of methods: static sensing and mobile sensing. The static sensing methods install sensors, such as vehicle detectors (VD) [1-2], infrared sensors and video image processors, on the roadways to detect vehicles passing over and record the traffic speed and traffic flow. These stationary sensors need power lines to operate and communication link to transmit the collected traffic data to a Traffic Information Center (TIC) in a centralized system or to the vehicles that used Vehicular ad hoc networks (VANETs) directly in a decentralized system. When the TIC or a VANET-connected car receives traffic data from static sensors, it may aggregate and correct the data to produce more accurate, reliable and real-time traffic information before disseminating to the road users. However, static sensors are expensive to deploy and maintain. They are mainly deployed on the main roads because of economic reasons.

Another method to collect traffic information is using mobile sensing devices to collect the floating car data (FCD). In general, FCD fall into two categories: floating cellular data and GPS-based probe cars. The principle of floating cellular data is to locate vehicles from the signals of mobile stations (MSs) constantly registering their locations to the cellular networks. When an MS is on a moving vehicle, cellular network control messages, such as location update and handover, would be exchanged between the MS and the network, and we can use two consecutive signals to detect vehicle speed and congestion on the road [9]. In this way, the mobile phones on moving vehicle can be used as probes. This approach is referred to as Cellular Floating Vehicle Data (CFVD). The second category uses probe cars equipped with Global Position System (GPS) and wireless communication capabilities to collect real-time traffic data, such as vehicle location, speed, and acceleration. Probe cars usually transmit these GPS data to the TIC or other probe cars periodically or when passing pre-determined locations. The information collected by the TIC or the on-board device of probe cars can infer current road condition. Compared with the traditional stationary sensing techniques described above, mobile sensing is more cost-effective and has a wider coverage of sensing area. There is no requirement of additional deployment of sensing devices along the roads and drivers can obtain traffic information from the TIC or VANET-connected probe cars. Service providers, such as TomTom [1], IntelliOne [4], ITIS Holdings plc [5] and Mediamobile [6], have developed applications based on FCD to provide real-time traffic information recently.

1.2 Motivation

Since mobile phones are widely used and the base stations of the cellular network system are deployed everywhere, CFVD have much wider coverage of sensing area and richer traffic data samples than the traditional GPS-equipped probe vehicles. However, the low accuracy in positioning an MS in the cellular network is the critical weakness since the network can only locate a mobile phone in an area of a cell, whose diameter is typically hundreds of meters. This position inaccuracy may result in inaccurate traffic information generated by CFVD. Compared with CFVD described above, the location of a GPS-equipped probe car can be much more accurate. According to "GPS.GOV", an Official U.S. Government, publish [7], GPS location errors are within tens of meters. Therefore, traffic speed obtained by GPS-equipped probe vehicles are much more accurate than that from cellular control messages. The conventional GPS-based probe cars rely on on-board GPS device to collect GPS data, and thus the coverage on road networks were restricted by the number of probe cars. Currently, smart phones equipped with GPS receivers are much more popular. Using smart phones as probes becomes feasible, and it can increase the number of probes hugely. Mohan et al. [8] has proposed a method using mobile smartphones as probes to monitor the traffic conditions. In thesis, we choose to use the GPS approach.

The system structure of traffic information systems using FCD can be classified into two categories: centralized structure and decentralized structure. In a centralized structure, a TIC collects traffic data from probe cars, and then distributes the traffic information to road users. This structure enables road users to obtain the complete traffic conditions on a large urban area since it is reliable client-server architecture. However, a centralized structure may experience a high delay (typically in the range of 20-50 minutes) [9] in disseminating traffic information because the center server need to receive and process a huge number of traffic raw data. Wischhof et al. [9] have proposed a decentralized traffic information system where probe cars exchange traffic data based on inter-vehicle communications. Each probe car periodically broadcasts the collected traffic data. As a result, a probe car can receive traffic information in nearby areas rapidly, and there is no bottleneck problem in a decentralized structure.

Considered a distributed system, each probe car must process traffic data that are received or collected, and the computing device, such as smart phone and on-board computers, is usually restricted by memory capacity, power storage and computing capability. Therefore, the data processing and dissemination module implemented in every node must be economical in consumption of memory, power, and computation. In addition, the limited bandwidth and car-to-car connection reliability are important problems to be addressed. In brief, using limited bandwidth and hardware resource to save travel time effectively is our primary issue.

1.3 Objective

As we described above, we intend to build a scalable and effective navigation system to save the travel time for road users. For the purpose of addressing scalability problem in a decentralized structure, we choose a utility-based gossip model CATE [10] as our dissemination model because gossip-based ad hoc routing is very effective [11] in disseminating large amounts of information in densely traffic. We build a complete simulation environment to simulate the navigation system and further investigate the relations between travel time and various parameters, such as the size of sending buffer, the broadcast period and the penetration rate. We also notice that the traffic congestions are frequently caused by intersection delay when cars wait to take a left turn. Therefore, we provide the turning delay in our design. Our traffic notification and navigation system has the following features:

(1) Probe cars are equipped with GPS receiver and VANET capability.

(2) Probe cars can dynamically reroute based on the collected traffic information.

(3) Probe cars periodically broadcast traffic data based on a utility function.

(4) The left turn delays at intersections are collected and disseminated.

(5) Probe cars have a fair and light computing overhead.

1.4 Summary

The remaining part is organized as follows. Chapter 2 describes the current work in Floating Car Data related to our system. Chapter 3 describes our system design in details. Chapter 4 discusses the experiment results and analysis in our system. Finally, we give our conclusions in Chapter 5.

Chapter 2 Background and Related Work

2.1 Distributed/Decentralized Traffic Information System

The Distributed/Decentralized Traffic Information System or calling Self-organizing Traffic Information System (SOTIS) consists of a group of vehicles that each member is a probe car and a traffic information center simultaneously. Every car collects locally traffic information and exchanges data with other cars through Inter-Vehicle Communication (IVC). IVC usually achieve by wireless connection such as IEEE 802.11p Short Range Communication (DSRC) / Wireless for Vehicular Environments (WAVE) radios and IEEE 802.11b Wi-Fi. Compared with centralized traffic information system, Decentralized Traffic Information System distributes traffic information to nearby cars with relatively lower delay. In conventional systems, probe cars periodically broadcast their traffic sample such as speed and travel time in a road segment to neighboring vehicles, and the vehicles which receive the broadcasted messages store and rebroadcast the traffic information to other. This propagation method is calling flooding. The flooding method makes a huge waste of bandwidth. Therefore, some methods that improve bandwidth efficiency were purposed.

A part of methods are using data aggregation technique to reduce redundant data and improve communication efficiency. C. Lochert et al. [12] purpose a hierarchical aggregation using multi-level landmarks on the hierarchical road network. The lowest level landmark may be a junction in the full road network. When a vehicle passes a road segment, it can records traffic information such as travel time and distributes them within the nearby area. Cars using the lower level traffic information to calculate the travel time between two landmarks on the higher level and distribute them to the further area. The detailed aggregation method describe as follow. First step is finding all possible routes between the two high-level landmarks. These routes are crossing lower-level landmarks. The travel times between these landmarks on the next lower level can be obtained by vehicle's collection, receiving from other car and using lower-level information to execute the aggregation algorithm. And final step is summarizing these routes travel time and choosing the minimal travel time to be the aggregated travel time between the two high-level landmarks. Fig 2.1 can depict the aggregation method described above

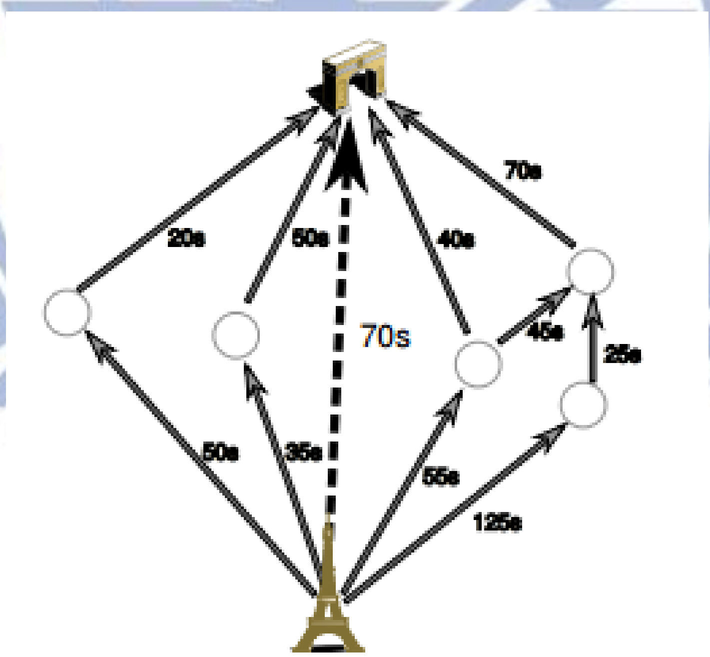


Fig. 2-1 Landmark aggregation [12]

While a driver is further away from both landmarks, he can get the aggregated traffic information between these two landmarks. When he comes closer to the respective area, he can receive more locally detailed traffic information and this information can help driver to choose an optimal route. This method can save the bandwidth by aggregation. However, aggregation may have traffic condition estimated error and the authors do not use a realistic navigation simulation to get the trustworthy evaluation.

Joon Ahn et al. [13] used the "belief value" and a novel time-decay sequential hypothesis testing (TD-SHT) in an in-network aggregation architecture to build an event report system, called Road Information Sharing Architecture (RISA). By using aggregation method, the bandwidth and storage used in relaying event reports can be saved. However, RISA has an inevitable problem. It may experience long aggregation delay and persistence error. The aggregation delay is caused by a time delay before a total belief value of an event is larger than the threshold. Conversely, the persistence error is caused by a time delay before a total belief value of an event is smaller than the minimal belief value. Fig2.2 depicts this problem in the aggregation mechanism.

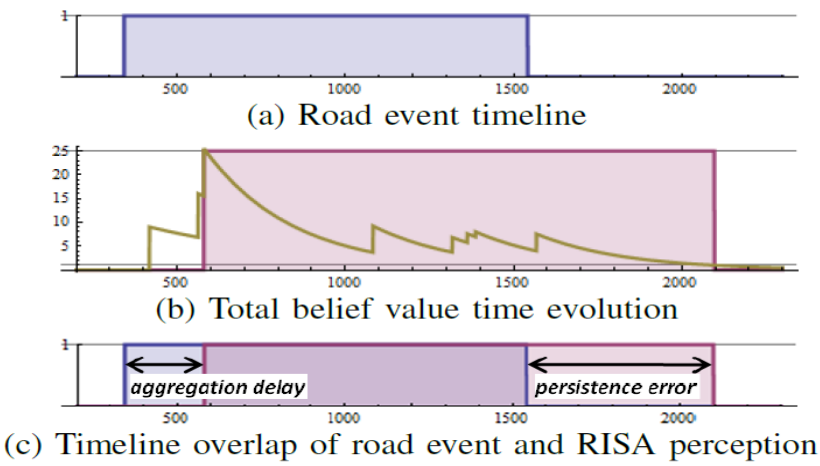


Fig. 2-2 Belief value interpretation [13]

The disadvantage of many aggregation methods is the aggregation delay because the aggregation function needs enough data samples to obtain a result, such as traffic state, with confidence. As a result, the time delay of data collection is inevitable, or the aggregated result may have a large accumulated error. In addition, the two systems described above also have a dissemination reliability problem. When the traffic density is low, the communications of wireless-equipped cars would be disconnected frequently. This makes real-time information dissemination very difficult. Deploying the infrastructure devices-supporting unit (SU), which is a stationary device performing the same data propagation function as the probe vehicles to mitigate the communication disconnection problem. But it has additional deployment cost although the optimal SU-placement method [12] can be used.

Several distributed systems use clustering techniques and cluster-to-cluster communications to improve data dissemination reliability and efficiency. Wegener et al. [14] use the hovering data clouds (HDC) [15], organic information complex and data dissemination protocol AutoCast [16] to design an event report system. The HDC is used to maintain traffic phenomenon information (e.g. car speeds become lower and lower). The principle of HDC is grouping vehicles around a traffic phenomenon into a cluster. A cluster is a HDC. In the traffic network, there may be several HDCs, and they maintain the traffic phenomenon information in the corresponding area. In the HDC, vehicles exchange information (intra-HDC communication) to determine this HDC's range. Outside an HDC, vehicles receive phenomenon information from the HDC (inter-HDC communication) and check whether they enter the HDC's area. When the phenomenon areas are moved, the related HDCs change dynamically. This dynamical clustering method can detect and hold important traffic information very well in the network. Project AutoNomos uses Organic Information Complex to maintain HDCs and recognize traffic events, as depicted in Fig2.3. The dissemination protocol AutoCast uses bandwidth efficiently by adapting broadcasting protocol for different vehicle density. When the network is dense, probabilistic flooding is used. Otherwise, when the network is sparse, conventional flooding is used. However, AutoNomos may have long aggregation delay because it aggregate large amount of data into a traffic event.

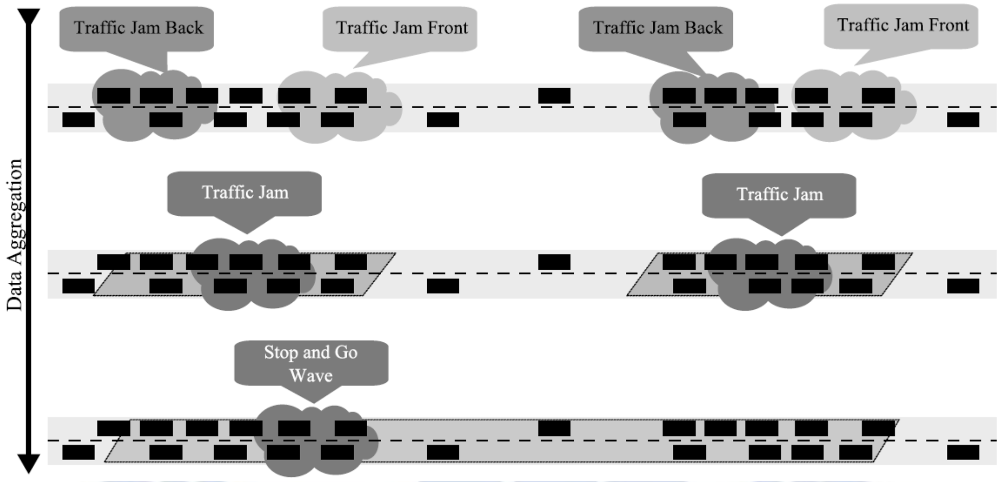


Fig. 2-3 Data aggregation to recognize traffic jam [16]

Y.-C. Chu et al. [17] use a Mobility-Adaptive clustering protocol that enables Vehicle-Adaptive cluster-to-cluster multi-hop forwarding to increase propagation reliability. The clustering algorithm uses two factors, movement prediction (MP) and re-clustering cost (RC), to dynamically choose cluster head (CH) and cluster tail (CT). These two factors can obtain by following equations (2.1 - 2.3):

$$MP_{i,j} = \begin{cases} \frac{S_{i,j}(\Delta t)}{R}, & S_{i,j} \leq R \\ 0, & (\Delta t) > R \end{cases} \qquad (2.1)$$

MPi,j is the movement prediction (MP) factor used to calculate the maximum propagation distance between vehicle j and vehicle i within Δt. $S_{i,j}(\Delta t)$ is the prediction distance between vehicle i and vehicle j within Δt. The formula: $S_{i,j}(\Delta t) = (S_{i,j} + V_j * \Delta t + 0.5 * (a_j * \Delta t^2)) - (S_{i,i} + V_i * \Delta t + 0.5 * (a_i * \Delta t^2))$. The a_i and a_j are acceleration of vehicle i and vehicle j. The V_i and V_j is the speed of vehicle i and vehicle j. And R is the communication range. In the other words, the distance between the CH and other members in this cluster is proportional to MP. Therefore, when MP has a larger value, it means that the cluster has larger coverage and the overlap network areas of all clusters on the road network are less but the probability of re-clustering is larger.

$$RC_{i,j} = \begin{cases} \frac{K-Z_j}{K}, Z_j < K \; and \; ((K - Z_j) \geq 0.5 * Z_i) \\ 0, (Z_j \geq K) or \; ((K - Z_j) < 0.5 * Z_i) \end{cases} \quad (2.2)$$

$RC_{i,j}$ is the re-clustering cost (RC) of the CH, vehicle i joining the cluster j. K is predefined cluster size, and Z_i is the cluster size of CH, vehicle i. Z_j is the cluster size of CH, vehicle j. This equation means that a cluster has a more proper size and is more stable when RC is higher. When MRF = (1-w)*MP + w*RC have a larger value, the adaptive degree of the CH is higher. The CH receives data from other cluster and broadcasts to other members in its cluster, and CT aggregates received data and sends to other cluster. Note that some clusters may have an overlap network area and hold redundant traffic information. MP (movement prediction) factor already mitigates this problem, and this problem can be solved by aggregation and propagation rules. In addition, the Vehicle-Adaptive cluster-to-cluster multi-hop forwarding method also use the movement prediction (MP) function to predict the distance between two clusters.MP uses the CH or CT of a cluster to calculate this distance. If a number of receivers have the same heading with a sender, the sender chooses the car that has largest distance as the relay node. Otherwise, it chooses the car that has the shortest distance as relay node. This inter-cluster communication method can propagates data widely and fast. To summarize this method, it has respectable dissemination reliability and it can reduce the usage of network bandwidth by propagating less traffic information. However, the traffic condition estimation errors occur because the disseminated traffic information may be incomplete and the MP function is too simple to use in urban road network.

S.-L. Tsao et al. [18] present a Two-Tier Traffic Information System that uses a clustering algorithm to couple the application-layer peer-to-peer (P2P) protocols on the high-tier P2P overlay and a broadcasting protocol, such as flooding, on the low-tier vehicular ad hoc network (VANET) overlay. This method focuses on increasing the information lookup success rate and decreasing the lookup latency. It is using clustering algorithms, such as the Lowest-ID and Highest-Degree algorithms, to group vehicles into several clusters and choose the superpeers in these clusters. These superpeers can form a high-tier P2P overlay through infrastructure wireless communication. Inside the cluster, cars exchange traffic information by flooding, while inter-cluster communications rely on the superpeers. Fig 2.4 depicts the system architecture. This system has both the advantages of VANET and infrastructure-based peer-to-peer network; it can provide fast and reliable data dissemination. Kowen Lu et al. [19] purpose an Adaptive Routing Algorithm to decrease the lookup latency further. It uses a connectivity module [20] to estimate the connectivity (that denotes that data dissemination can succeed) probability of road segments. If the connectivity probability is higher than a predefined threshold, the inter-cluster communications rely on the low-tier VANET overlay. Otherwise, the inter-cluster communications rely on a high-tier P2P overlay. This method has respectable dissemination speed and it has more dissemination reliability than conventional VANET-based systems. However, when in a dense network, data dissemination may be decayed like flooding methods and it also has additional costs to maintain the clusters.

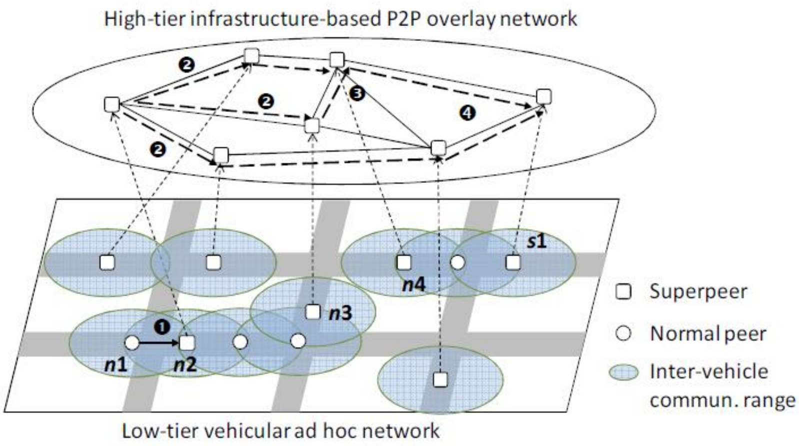


Fig. 2-4 Routing in two-tier traffic information system

The distributed systems that use clustering methods have to deal with the scalability problem. Each car must consume additional bandwidth for exchanging control signals and computation power to maintain the clusters and the routing path of cluster-to-cluster communications, especially for the cluster leader that has more overhead than other members in the cluster. In a dense network, the cars require a lot of bandwidth to support the work. Therefore, the limited bandwidth restricts the scalability in the systems. In addition, many distributed systems that use clustering methods also use aggregation methods, such as AutoNomos and the method proposed by Y.-C. Chu et al.. Therefore, aggregation problems, such as traffic condition estimation error and data aggregation delay also exist in these systems.

C. Sommer et al. [21] purpose a method based on dynamical broadcasting period. This method uses two factors to control broadcasting period: channel quality and data utility. When the communication channel quality is poor and the data utility is high, the shorter broadcasting period is used. Conversely, if the communication channel quality is good and the data utility is low, a longer broadcasting period is used. In other words, a car has a shorter broadcasting period in a higher dense traffic flow because channel quality is usually poor in this traffic situation. This method may propagate data with a longer distance and has a faster propagation speed if the period calculation function is reliable. However, the authors only focus on communication reliability, and they do not test the performance of saving travel time for road users.

Z. J. Haas et al. [11] purpose a gossip-based Ad Hoc Routing protocol, the main idea is probabilistic flooding. In contrast with the methods previously described, this protocol has very good dissemination efficiency in dense traffic. I. Leontiadis et al. [10] purpose a utility-based gossip model that is used in their navigation system CATE to improve dissemination efficiency. It uses a utility function to choose more important data samples in a vehicle, and then it broadcasts these data samples to others. This method can hold the important traffic information in the network and save much more bandwidth by the utility function. In addition, it can receive the necessary data from other cars in real time. Unlike previous methods that lack a realistic experiment environment, CATE has been proved to have a great performance and scalability by the realistic experiment data. However, there are some deficiencies that need to be improved; for example, intersection delays are not considered in route planning and secondary congestion may occur when too many cars follow the advice of the navigation system. Because of the reasons previously described, we choose the utility-based gossip model to build our system and improve the performance of the system.

2.2 SUMO

Simulation of Urban Mobility (SUMO) is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks [22-23]. When we want to improve traffic condition by developing a related system, a valid simulation model to work with is needed because of economic reasons. SUMO can help us to build a complete and realistic traffic simulation environment by using its road network and traffic flow editor. The road network editor uses custom nodes and edges to build a road network, and we can also flexibly modify the details of the road network, such as specifying the cycle of traffic signals and internal edge connections. In addition, it can import a variety of open source digital maps, such as OpenStreetMap. The traffic flow editor can specify not only the origins and destinations (O/D) but also the routing paths for the traffic flows. It also supports Statistics data, such as population, population's age brackets, work hours and work position distribution to produce the realistic traffic flows in the urban. Fig2.4 is a simulation example of SUMO.

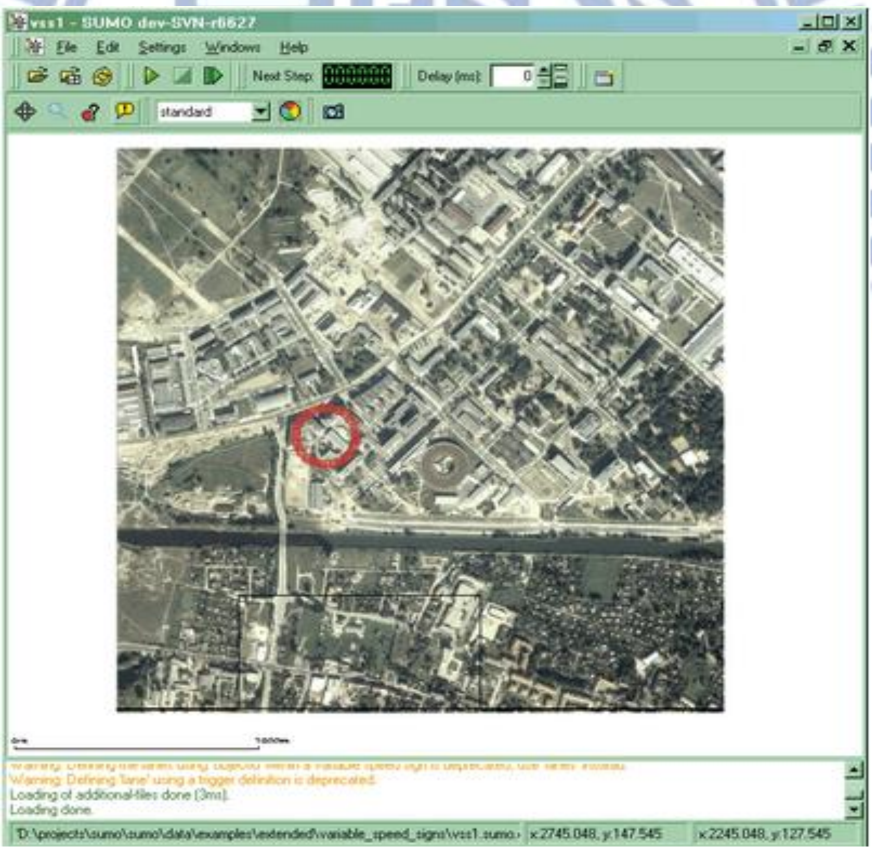


Fig. 2-5 A simulation example of SUMO. [22]

SUMO supports both command-line interface and GUI interface for running simulation, and the important reason for using SUMO is that, it can easily be coupled with network simulator by using traCI [24] because using traCI can enter commands to change each car's driving behavior and traffic control signal on simulation running time. It can help us to build a realistic simulation for a navigation system.

2.3 OMNeT++

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators [25]. The INET framework, which contains models for a variety of protocols used in the network, such as wired and wireless link layer protocols (Ethernet, IEEE802.11, etc.), support for mobility, MANET protocols and many other protocols and components. By using the Network Description (NED) file which can be modeled graphically to describe the basic modules' relationships and their communication links, we can design an extended IVC protocol. In addition, the trace-driven simulation can obtain more realistic evaluation of IVC protocols by using the realistic trace data, such as observations of real traffic or productions of road traffic micro-simulation tools to represent node mobility patterns. We can use this event-based network simulator to simulate the VANET. Furthermore, using Veins described below to couple OMNeT++ with SUMO, we can have a more realistic experiment environment by updating traces on simulation running time.

2.4 VEINS—Vehicles in Network Simulation

In a navigation system based on a distributed traffic information system built in the vehicular ad hoc network (VANET), each car that uses this navigation system collects traffic information and exchange information with other, and then the car uses this traffic information to change its travel path for saving travel time. Therefore, if we want to improve this navigation system, the realistic simulation environment for the system is needed, because it is very helpful to produce trustworthy experiment data and we can use this data to effectively improve the performance of the system through data analysis. In this thesis, we choose Veins described below to achieve the realistic simulation for our system. Fig 2.5 shows the architecture of Veins.

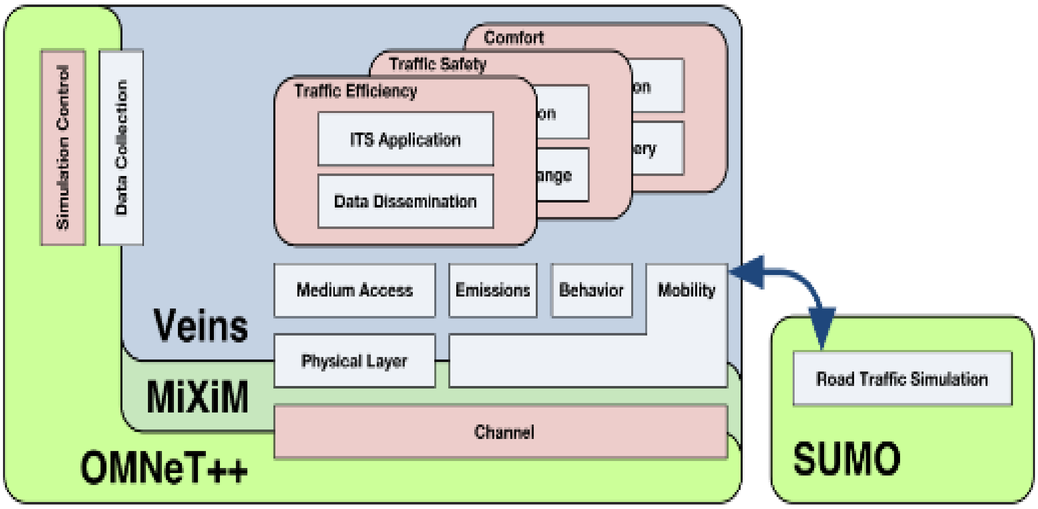


Fig. 2-6 The architecture of Veins. [26]

Veins is an open source framework for running vehicular network simulations [26-27]. It bi-directionally coupled OMNeT++ and SUMO together by using a timing-trigger manner and a simple request/response protocol. OMNeT++ and SUMO guarantee synchronous execution at defined intervals by buffering all commands arriving between two timesteps. At each timestep, OMNeT++ send the buffered commands to SUMO for triggering the corresponding traffic road simulation, and then SUMO return the simulation results, such as every vehicle's position for advancing network simulation. Fig. 2.6 shows the control modules integrated with OMNeT++ and SUMO. Fig. 2.7 shows the interaction between both simulators in the form of a message sequence chart. Fig 2.8 is an example of simulation by Veins that uses an IVC protocol for communications.

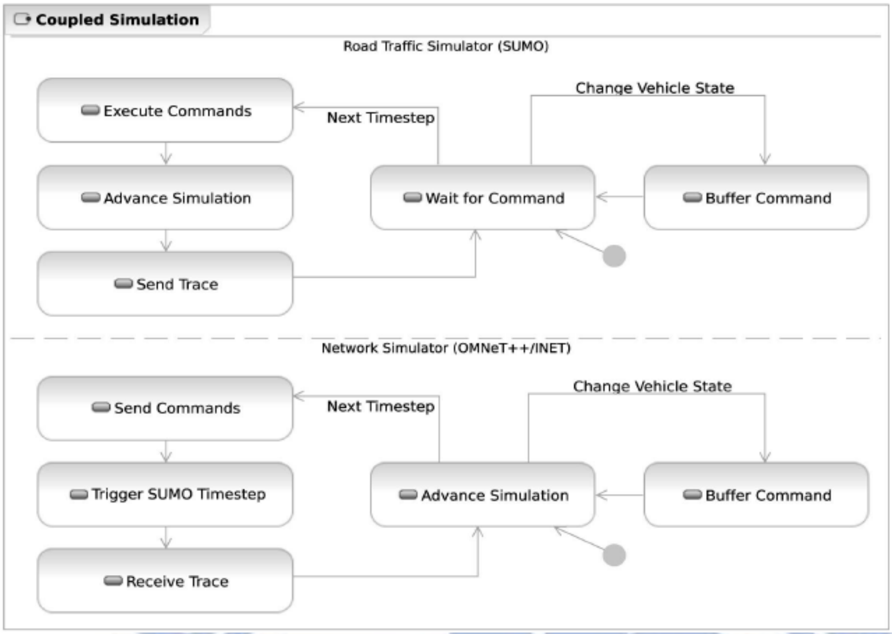


Fig. 2-7 Overview of the coupled simulation framework. State machines of road traffic and network simulator communication modules. [27]

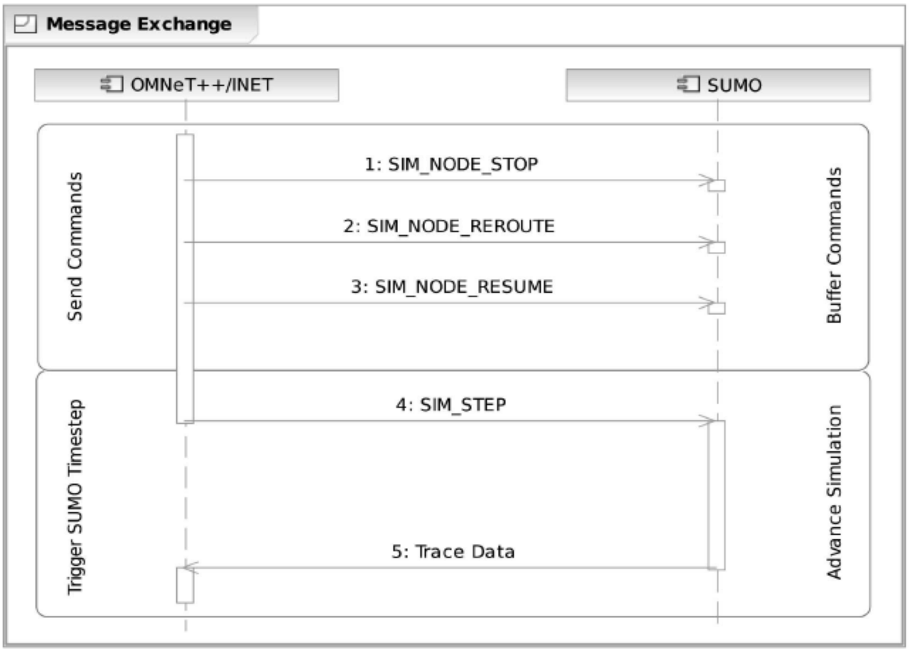


Fig. 2-8 Sequence diagram of messages exchanged between network and road traffic simulator communication modules. Command execution is delayed until the next road traffic simulation timestep is triggered. [27]

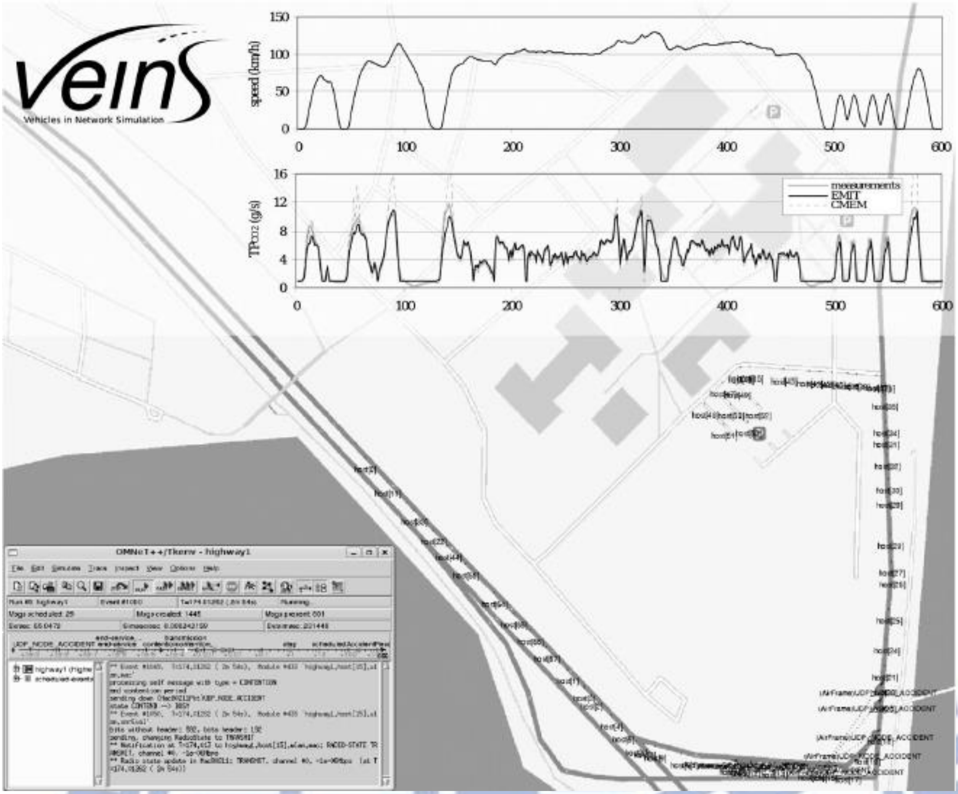


Fig. 2-9 Bidirectionally coupled network and road traffic simulation.Shown are vehicles controlled by Veins that use an IVC protocol for communication. [27]

Chapter 3 The System Design

3.1 System Overview

Our system uses a decentralized architecture that consists of vehicles which are inter-connected with a vehicular ad hoc network (VANET). Fig 3-1 depicts our system architecture. The detailed functions of each component will be described below.

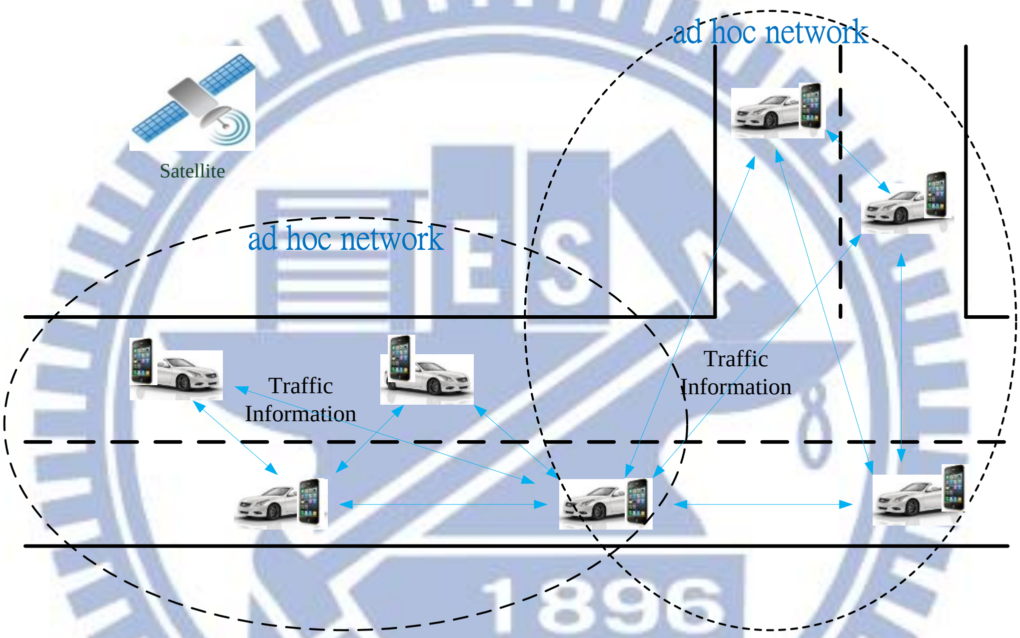


Fig. 3-1 The System Architecture

The goal of our navigation system is to disseminate sufficient real-rime traffic information under a limited communication bandwidth to probe cars equipped with GPS receivers and VANET capability. Based on the received traffic information, an on-board navigation system estimates the traffic conditions and instructs the driver to take a route that minimizes the travel time. In addition, even if only a part of vehicles use our system, the navigation-embedded probe cars still can avoid the congested road segment, and the congestion will be mitigated. Therefore, the traffic conditions in the whole road network can also be improved.

In this system, each probe vehicle not only obtains traffic information from its mobile sensors but also shares the traffic information with neighboring vehicles via an ad hoc manner. Periodically, vehicles update the current traffic condition from the collected information and then re-compute the best travel route from the current position to destination. Therefore, our navigation system should be embedded in each car. Our navigation system, which is an extension of CATE model [10], consists of a dissemination module that propagates data samples under the limited bandwidth, a traffic condition estimator that uses the received or collected data samples to assess the current traffic condition, and a navigation module that utilizes the digital map and current traffic condition to perform the route planning algorithm.

We assume that each probe vehicle carries a smartphone equipped with a GPS receiver and VANET communication capability. In addition, the smartphone also has a digital map in its memory. Since the travel time of each road segment is an effective observation to represent the degree of congestion, each vehicle collects the travel times of the road segments as data samples when it passes these street sections. A traffic sample forms as {**linkID, delay, timestamp, carID**}, which is same with CATE. "**linkID**" is the unique ID which is used to identify the road segment. Similarly, the **carID** represents the producer of this sample. The **delay** field records the travel time. The **timestamp** is the GPS time at which the sample was collected, i.e., the time when the vehicle exits the observed road section. Each probe car periodically records its position; the interval is one second. By coupling the recorded locations with the digital map, each car can use the two GPS timestamps that are the nearest to the start point and the end point of a road segment to calculate the travel time when it passes the road segment and the estimation error of the travel time can be limited within one second because the interval of GPS position updating is short enough. Fig. 3-2 shows this navigation system architecture in each car.

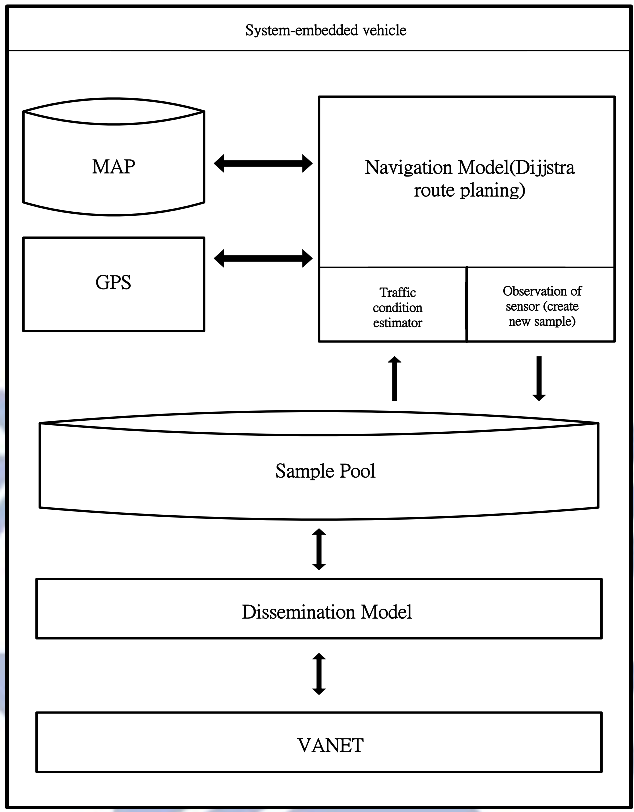


Fig. 3-2 Showing this navigation system architecture in each car

3.2 CATE Module

To make it easier for readers to understand our system, we first describe the original CATE module and how to build it in this section.

3.2.1 Dissemination Module

For the purpose of utilizing the bandwidth effectively, we should choose more important information to broadcast first. Therefore, CATE selects a subset of the data samples that are valid in the storage containing the received data samples and the samples observed by self by a utility function [10] and broadcast them to the neighboring cars. When a car receives the broadcasted information, it integrates the own samples with the received samples and propagate them at the next broadcast time. In our implementation, duplicated information is discarded when cars integrate these data samples. This can reduce the number of rebroadcasted data samples and further improves the bandwidth efficiency.

The core of this dissemination module is the utility function which estimates the effectiveness of each sample. CATE uses this function to prioritize the data samples stored in the car and rank them by the priority. Subsequently, the K (which is predefined based on sending buffer size) road segments with the highest utility are broadcasted to the neighbors. The utility function considers two factors: data freshness ($r_{mostRecent}$) and the coverage of data spread ($r_{lessbroadcasted}$). Let current time denote the current time stamp, and $timestamp_{linkID}$ the sample's collected time. NB_{linkID} denotes the number of received samples about this road segment during the past **t** minutes.

1. Data freshness: Prioritize the road segments based on how recent the information is, and the more recent samples have the higher priorities. We can define Equation 3.1 based on $timestamp_{linkID}$ to achieve this.

$$r_{mostRecent} = \frac{1}{(current\ time - timestamp_{linkID})} \tag{3.1}$$

If $current\ time = timestamp_{linkID}$, we set the maximal priority to this link.

2. The coverage of data spread: Prioritize the road segments based on how sparse the disseminated information is, and the more sparse samples have the higher priorities. CATE uses Equation 3.2 which is inverse proportion to NB_{linkID} to achieve this.

$$r_{lessbroadcasted} = \frac{1}{NB_{linkID}} \tag{3.2}$$

If NB_{linkID} = 0, we set the maximal priority to this road segment.

Finally, the utility of a road segment: $U_{link} = r_{mostRecent} + r_{lessbroadcasted}$. In this thesis, we set the time window t = 1 min. In this function, $r_{mostRecent}$ can help to pick the real-time information, and $r_{lessbroadcasted}$ can not only enhance high geographical coverage of the disseminated information but also avoid similar information of the same road segment re-broadcasted in the network. Fig. 3-3 illustrates the flow char of this dissemination module.

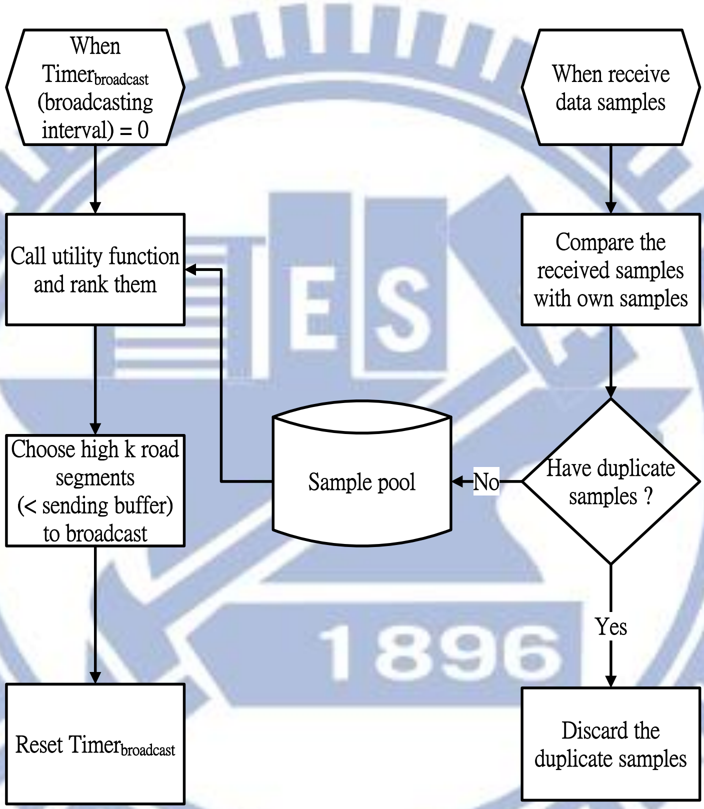


Fig. 3-3 Showing the flow char of this dissemination module.

3.2.2 Traffic Estimation and Dynamic Routing Module

As we described previously, each vehicle obtains a set of data samples through Inter-Vehicle-Communications (IVC). The important issue is how to estimate the current traffic condition and reroute the vehicle dynamically.

We choose the Dijkstra shortest path algorithm as our route planning algorithm to periodically re-compute the best route of the vehicle based on the current traffic condition. In practice, the edge weight of each road segment is the updated road segment delay. It means the shortest path which is chosen by Dijkstra has the lowest travel time.

The traffic condition estimator uses the collected data sample to periodically estimate and update the current traffic condition. To estimate an accurate traffic state is very difficult, because the traffic samples of a certain road segment may be too old to use or simply unavailable. However, these two problems can be mitigated by the dissemination module. CATE uses a simple estimation method [10] to further mitigate the two problems.

1. Default: Assume a road segment with a free speed. Segment's weight can be calculated by dividing its length by its speed limit as indicated in Equation 3.3.

   $$W_{linkID} = \frac{LinkLength}{SpeedLimit} \quad (3.3)$$

   If there is no information about a road segment in the car's storage, set the default weight for this road segment. The reason is that this situation usually indicates there are sparse vehicles on the road segment.

2. Most recent estimate: Our system selects the most recent sample for each road segment indicated in **timestamp** as the traffic condition, as indicated in Equation 3.4.

   $$W_{linkID} = Delay_{linkID,mostRecentSample} \quad (3.4)$$

   This method uses only the most recent sample to represent the current traffic condition.

CATE also purposes a Bays' estimate and a Bays' with aging estimate to assess the traffic condition, but the experiment results indicate that the most recent estimate provide the most satisfactory performance. Therefore, we choose the most recent estimate. If there are many malicious reports, the estimation accuracy would be low, but this is not within the scope of our discussion. Fig. 3-4 shows the flow chart of this traffic estimation and dynamic routing module.

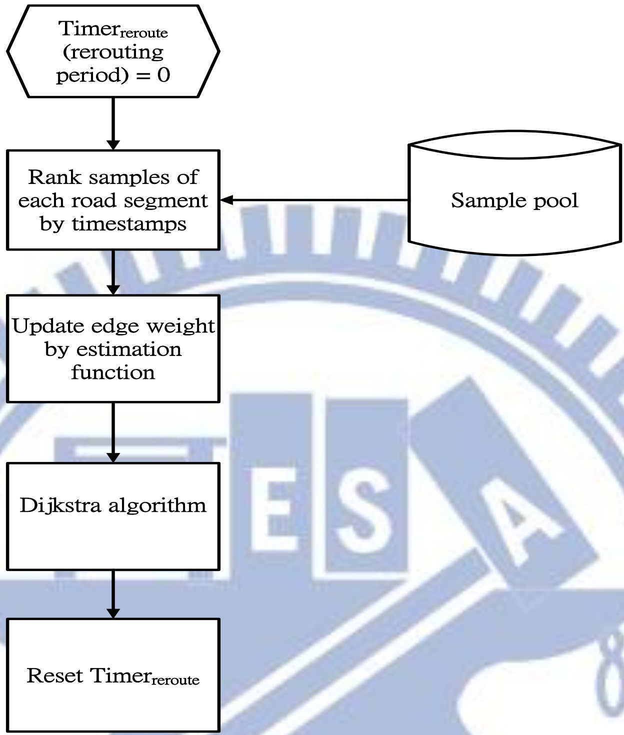


Fig. 3-4 Showing the flow char of this traffic estimation and dynamic routing module.

3.3 Provide the Turning Delay of Intersections

The intersection delay is one of primary origins of traffic congestion, and this delay is usually caused by slowing down to turn at the cross and traffic signal. In order to mitigate the influence of intersection delay, we have to know how to obtain this delay. There are many studies in estimating the intersection delay [28-31], and the dynamical traffic signal controlling can mitigate the traffic signal delay. In this thesis, we use a simple method to estimate the turning delay. Studying CATE's system, we learned that some vehicles wait for a long period of time, with respect to the travel time of the traveling road segment, to make a left turn at the intersection. The travel time of the road segment observed by left turn vehicles, and that of straight going vehicles differ by a large amount. As a result, it is difficult to determine the travel time of the road segment in CATE's system.

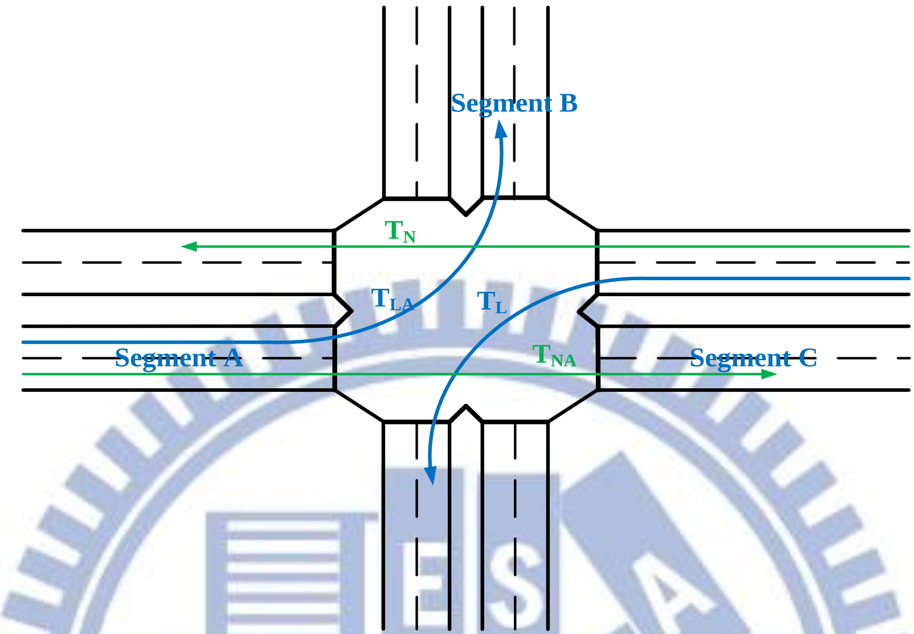


Fig. 3-5 Diagrams T_L and T_N

Because left turn cars need to yield to straight going cars, the waiting time of left turn cars at an intersection may cause a large time delay. Compared with the waiting delay, the turning delay that is caused by slowing down can be ignored. For the reasons described above and the limited bandwidth, we only consider the left turn delay because the right turn cars have no need to yield to the cars in the opposite direction. As Fig. 3-5 illustrates, T_{LA} is the travel time of **segmentA** that contains the left turn delay, and the T_{NA} is the travel time of straight going cars in **segmentA**, i.e., the cars enter **segmentC**. We assume the travel time of **segmentA** that contains the right turn delay approximates T_{NA}. Therefore, each road segment has two types of samples, $T_{LlinkID}$ and $T_{NlinkID}$. When a car enters a new road segment, it creates a sample for the previous segment and determines that the sample is T_L or T_N. We can use the original dissemination module to spread the data samples, but the traffic estimator should be modified, as in Equation 3.5.

$$TurnLeftDelay_{LlinkID} = \begin{cases} W_{LlinkID} - W_{NlinkID}, if\ W_{LlinkID} - W_{NlinkID} > 0 \\ 0, if\ W_{LlinkID} - W_{NlinkID} \leq 0 \end{cases} \quad (3.5)$$

First, we use the original estimation function to obtain $W_{LlinkID}$ and $W_{NlinkID}$, and then use the Equation 3.5 to obtain the left turn delay. The route planning algorithm uses these delays to further reduce the travel time. Of course, if there is no information of $W_{LlinkID}$ or $W_{NlinkID}$, the default value which is the travel time with a free speed is used. Fig. 3-6 shows the flow chart of the new traffic condition estimator cooperating with dynamic routing module.

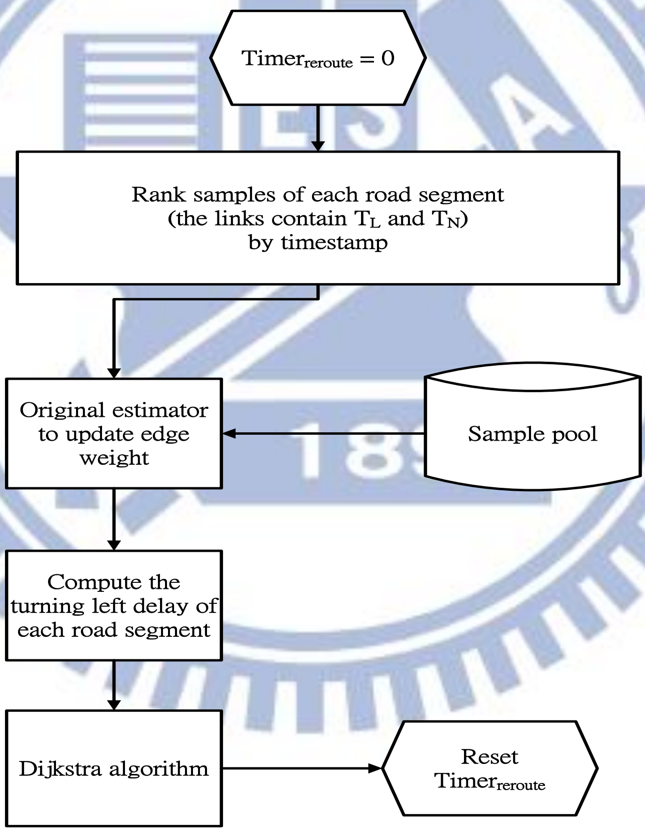


Fig. 3-6 Showing the flow char of the new traffic condition estimator cooperating with dynamic routing module

3.4 Implementation

In order to implement the navigation system, we use the Veins library to develop the application. First, we use the coupling program that developed by Python to couple the SUMO with OMNeT++. Then, using API ChangeRoute to notify SUMO to reroutes cars based on edge weights, and the setting message function is used by the dissemination module to set the broadcast content. Each car performs the same program that is written as a class. Each class maintains an individual table to store all traffic data, which is collected or received. In the road network simulation, each intersection has internal roads in it. When the left turn delay is not considered, each internal edge weight has default (free speed) weight. If the left turn delay is considered, the calculated turning delay will be added to corresponding internal edge weight as shown in Figure 3-7. For the purpose of optimizing the program performance, we modify this open source library: send all changing edge weight command to SUMO every second, and then send a change route command to execute the Dijkstra route planning based on updated edge weight every 30 seconds. Fig3-8 shows the SUMO and OMNeT++ intersections in our program.

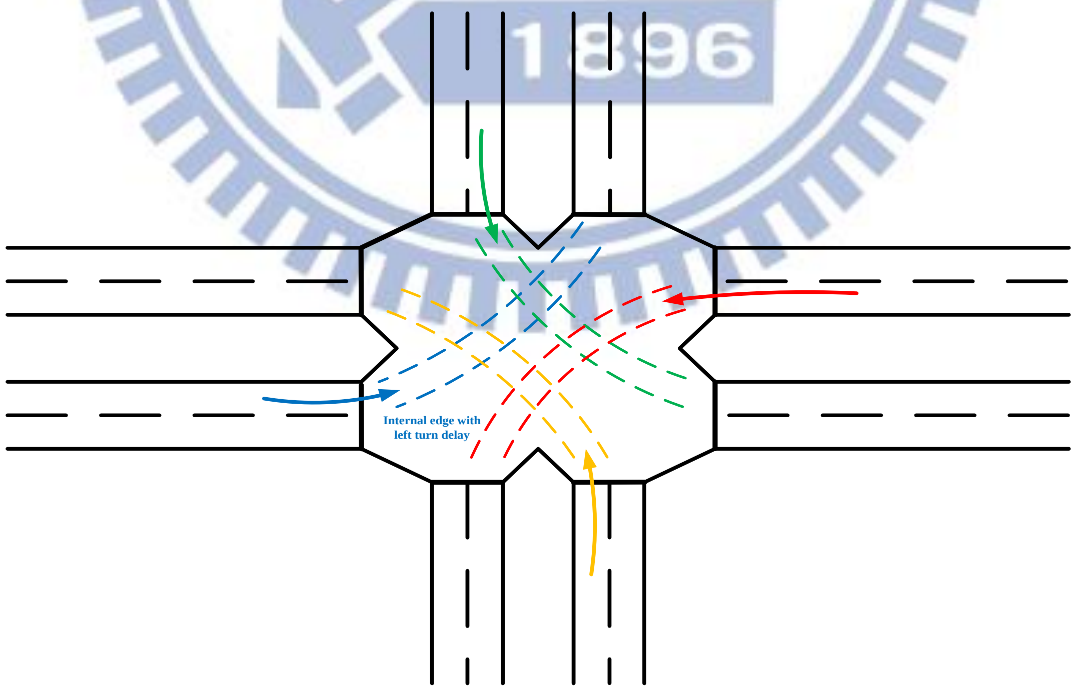


Fig. 3-7 Internal edge that placed the turning delay

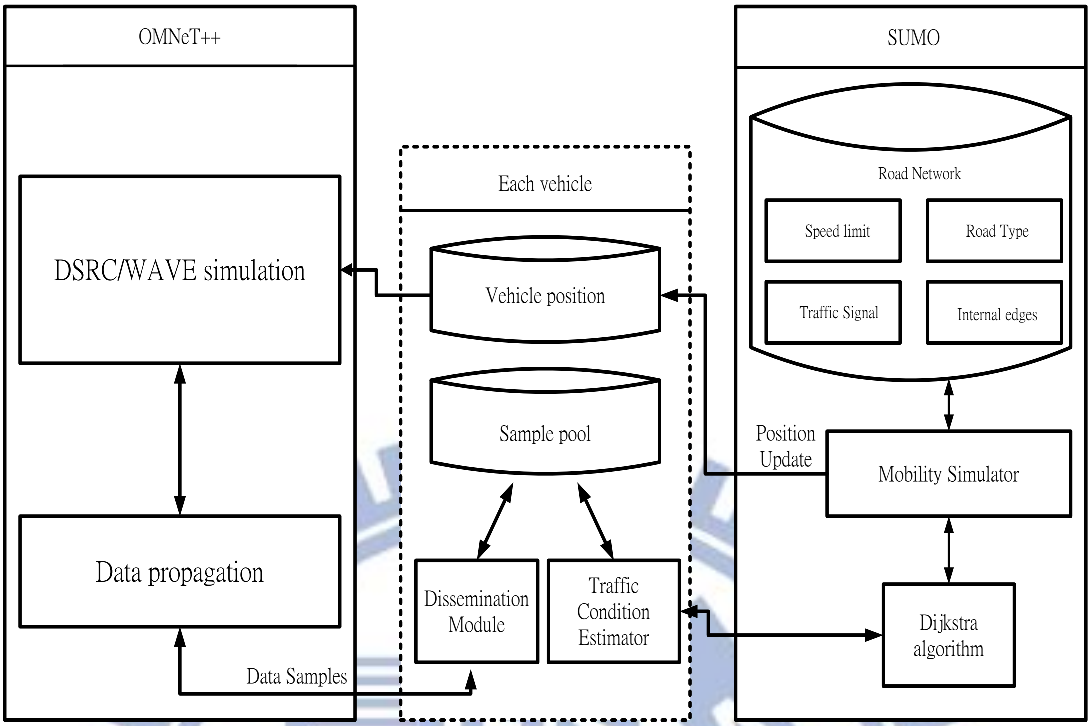


Fig. 3-8 SUMO and OMNeT++ intersections in our program.

Chapter 4 Evaluation

4.1 Simulation Environment

We use Veins to combine network simulator OMNeT++ with traffic simulator SUMO to develop a complete simulation for the navigation system. Fig. 4-1 indicates a 5 by 5 grid road network within a 3KM by 3KM area, which is built by SUMO. This road network consists of 120 road segments and 25 intersections with traffic signals. The length of each segment is 500 meters. In addition, there is a broadened lane about 50 meters extending from each intersection used by left turn vehicles only, as shown in Fig. 4-2.

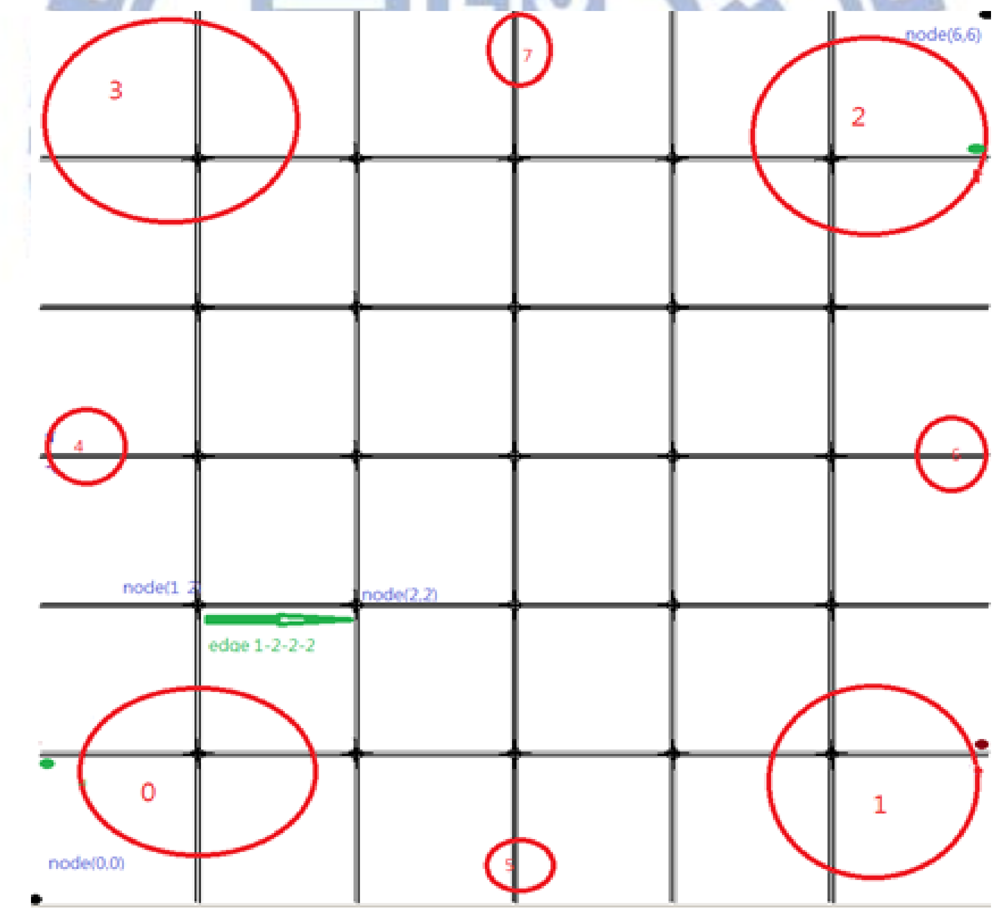


Fig. 4-1 5 by 5 grid road network within the 3KM by 3KM

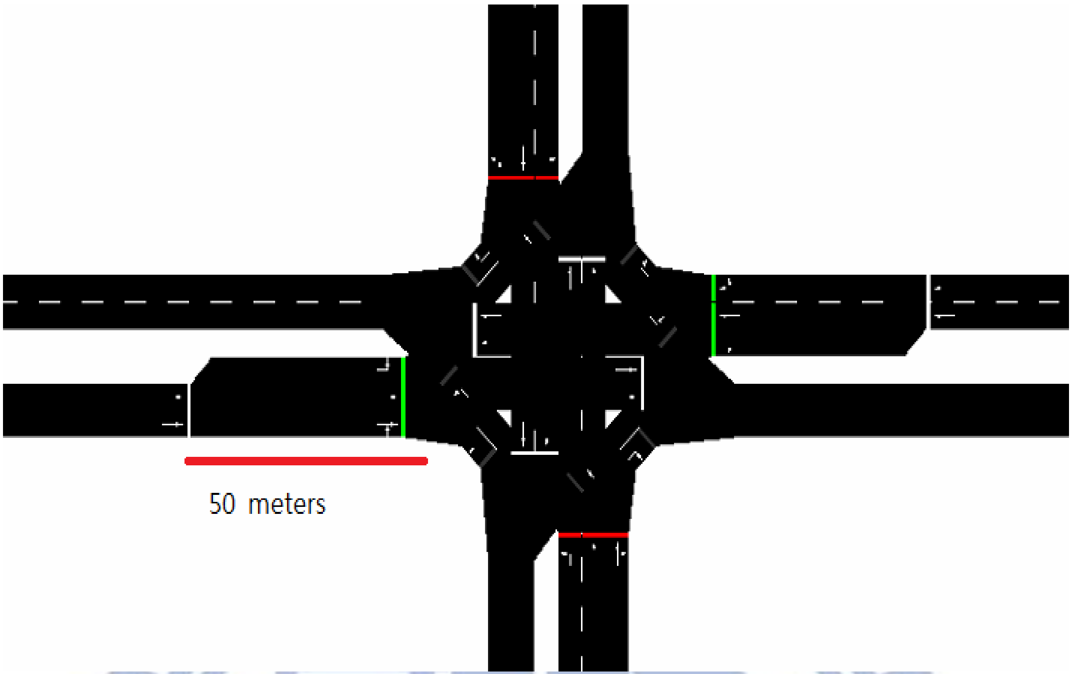


Fig. 4-2 Broadened lane of each intersection

The areas marked in red circles in Fig. 4-1 are traffic flow inputting areas. We can assign the Origin/Destination (O/D) segments to each flow, and the actual routes will be determined by SUMO. The name of each segment is based on the coordinates of the intersections. For example, the segment from intersection (0,1) to intersection (1,1) is denoted by edge0-1-1-1. It also means each road segment is directional, i.e., it has only one direction. In our simulation, we assign 12 flows, and each flow has two different destination edges. For example, flow0-0 is from edge0-1-1-1 (green point in area 0) to edge5-5-6-5 (green point in area 2) and edge5-1-6-1 (brown point in area 1), and each destination edge receives half of the flow. Table 4-1 lists the 12 flows' O/D pairs. Each input flows can be divided into 3 intervals during the simulation time: (1) 0-4500 sec: 288 cars/hour, low density traffic, (2) 4500-6300 sec: 600 cars/hour, high density traffic, (3) 6300~10800 sec: 288 cars/hour, low density traffic. Table 4-2 lists the flow input in 3 intervals. The total number of cars injected is 12240. Other detailed parameters of our road network simulation are listed in Table 4-3.

Table 4-1 The O/D pairs of each traffic flow

flow	Origin Edge	Destination Edge
Flow0-0	edge0-1-1-1	edge5-5-6-5
		edge5-1-6-1
Flow0-1	edge1-0-1-1	edge5-5-5-6
		edge1-5-1-6
Flow1-0	edge6-1-5-1	edge1-5-0-5
		edge1-1-0-1
Flow1-1	edge5-0-5-1	edge1-5-1-6
		edge5-5-5-6
Flow2-0	edge6-5-5-5	edge1-1-0-1
		edge1-5-0-5
Flow2-1	edge5-6-5-5	edge1-1-1-0
		edge5-1-5-0
Flow3-0	edge0-5-1-5	edge5-1-6-1
		edge5-5-6-5
Flow3-1	edge1-6-1-5	edge5-1-5-0
		edge1-1-1-0
Flow4-0	edge0-3-1-3	edge5-1-6-1
		edge5-5-6-5
Flow5-0	edge3-0-3-1	edge1-5-1-6
		edge5-5-5-6
Flow6-0	edge6-3-5-3	edge1-1-0-1
		edge1-5-0-5
Flow7-0	edge3-6-3-5	edge1-1-1-0
		edge5-1-5-0

Table 4-2 Input car number of each traffic flow

Time interval	Car number
0 ~ 4500 sec	288 cars/hour
4500 ~ 6300 sec	600 cars/hour
6300 ~ 10800 sec	288 cars/hour

Table 4-3 Parameters of road network simulation

Total road network area	x-axis: 3km y-axis: 3km		
Number of lanes	450meter: 2 lanes Near intersection about 50 meters: 3 lanes		
Number of segments	120		
Free Speed	mps:15 meter / sec Kmph: 54 km / hour		
Simulation time	3 hr		
Traffic light cycle(100 seconds)	Green	Yellow	Red
Intersection	45 s	5 s	50 s

The simulation of Vehicular Ad-Hoc Networks (VANET) is achieved by simulating the IEEE 802.11p mac1609.4 DSRC/WAVE. DSRC usually has 3-27 Mbps bitrate and 300-1000 meters of communication range. In our simulation, we conservatively set 18 Mbps bitrate and 250-meter communication range. Table 4-4 lists the detailed parameters of our communication network simulation.

Table 4-4 Parameters of communication network simulation

parameter	value
channelcontrol.pMax	5mW
channelcontrol.sat	-89dBm
channelcontrol.alpha	2.0
channelcontrol.carrierFrequency	5.890e9 Hz
nic.mac1609_4.txPower	20mW
nic.mac1609_4.bitrate	18Mbps
nic.phy80211p.sensitivity	-89dBm
nic.phy80211p.maxTXPower	10mW
nic.phy80211p.thermalNoise	-110dBm
nic.phy80211p.useThermalNoise	true
nic.phy80211p.usePropagationDelay	true

At the end of the simulation, Veins generates a scalars file containing the carID, start time, stop time and total travel time of each car. Therefore, we can easily use this file to obtain important measurements including the average travel time of the observed cars. We will use the average travel time as the main basis for evaluating the performance of the navigation system.

4.2 Evaluations

We utilize the C++ language to develop a program that implements the navigation application containing dissemination and rerouting in Veins, and use the Java language to develop a parsing program to retrieve measurements of each car from the recorded file at the end of simulation. In order to increase the reality and trustworthiness of the simulation data, we retrieve only data in simulation time

3600-7200 sec that contains low density flow and high density flow for analysis. Flow inputs in 0-3600 sec are the initial flows to fill the empty road network, and flow inputs in 7200-10800 sec are intended to keep the flows continually inputting to this network during observation time. There are 5328 cars being observed.

In the real world, we can't hope everyone uses the vehicle navigation system. Therefore, we intend to know how the penetration rate of probe cars affects the travel time. In addition, the size of sending buffer and broadcasting period also has influence on travel time. We will discuss the effects of broadcasting period on the travel time for different penetration rates. Subsequently, the effects of sending buffer size on the travel time for different penetration rates will be discussed. Finally, we will study the performance improvement when left turn delays are considered in the navigation system.

4.2.1 The Effects of Broadcasting Period

Broadcast period affects the data updating frequency. While the broadcasting interval become shorter, more messages will be spread in the network, and it also means more new samples may be received. Therefore, each car may have more fresh traffic information. In the following, we will test multiple broadcasting periods with different penetration rates, and observe the variation of the average travel time of all observed cars. The sending buffer size is fixed to 250 bytes because it is a threshold to approach optimal navigation performance, and the interval to perform rerouting is 30sec.

Fig. 4-3 depicts the effects of broadcasting period on the average travel time. The orange line, navy blue line, red line, green line, purple line and water blue line represent the 0% (no navigation), 15%, 25%, 35%, 45% and 100% (all use navigation) penetration rates, respectively. The x-axis is broadcasting interval and the y-axis is the average travel time. We can observe that the broadcasting period is positively correlated to the average travel time. With broadcasting period in 10-30 sec, the average travel time remain almost unchanged. With broadcasting period in 30-120 sec, the average travel time increases significantly as the broadcast period increases. We also notice that 15% and 25% penetration rates have approximately the same travel time when broadcasting period is 120 sec, and this travel time also approximates to that of 0% penetration rate. This means this navigation system loses efficacy in low penetration rate (15% - 25%) and long broadcasting interval (120 sec). It is worth mentioned that 45% penetration rate has approximate performance with 100% penetration rate when the broadcasting period is in 10-30 sec.

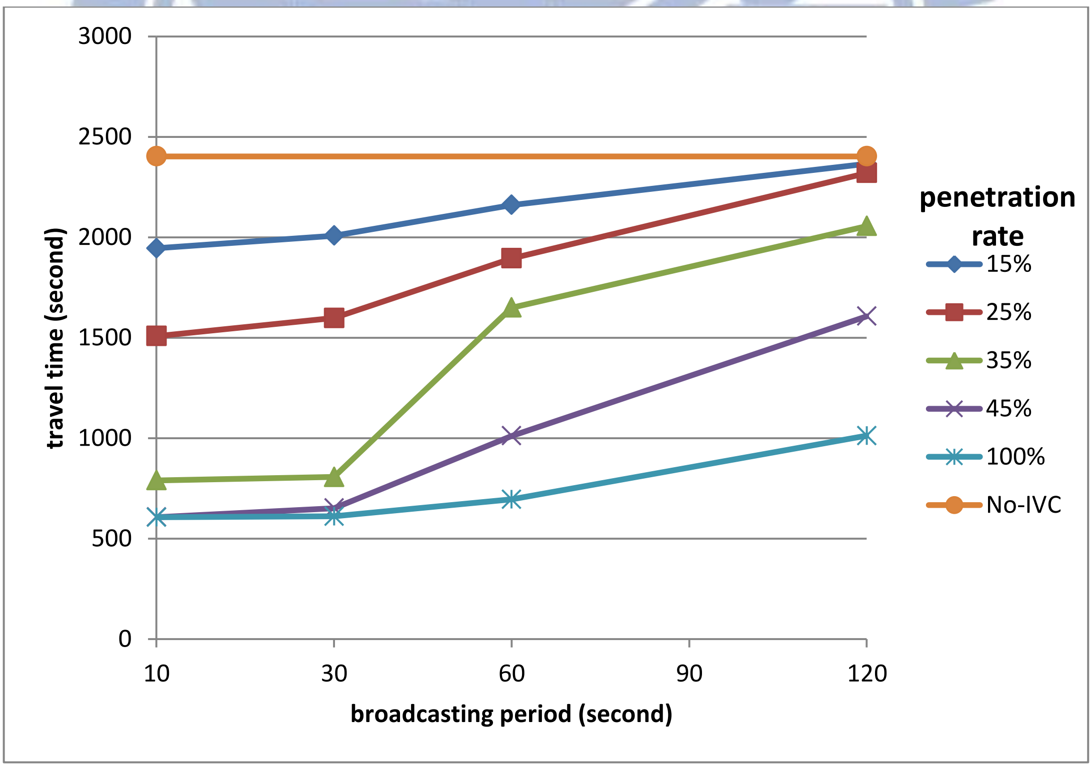


Fig. 4-3 The effects of broadcasting period on the average travel time

Fig. 4-4 depicts the effects of penetration rate on the average travel time with different broadcasting periods. The navy blue line, red line, green line and purple line represent the 10, 30, 60 and 120-sec broadcasting periods, respectively, and the water blue line no navigation. The results indicate that the average travel time decreases as the penetration rate increases for all broadcast periods. Note that the navy blue line (10 sec) is very close to the red line (30 sec). This means 10-sec broadcasting period has the average travel time approximating to that of 30-sec broadcasting period, especially in high penetration rates (35% - 100%). This phenomenon is caused by length of each segment and the vehicle's speed. Each segment has 500 meters length, and the free speed is 15 meters/sec. Every car produces a new sample at least 33 seconds. Therefore, the improvement of data freshness is limited when the broadcasting period is shorter than 30 sec.

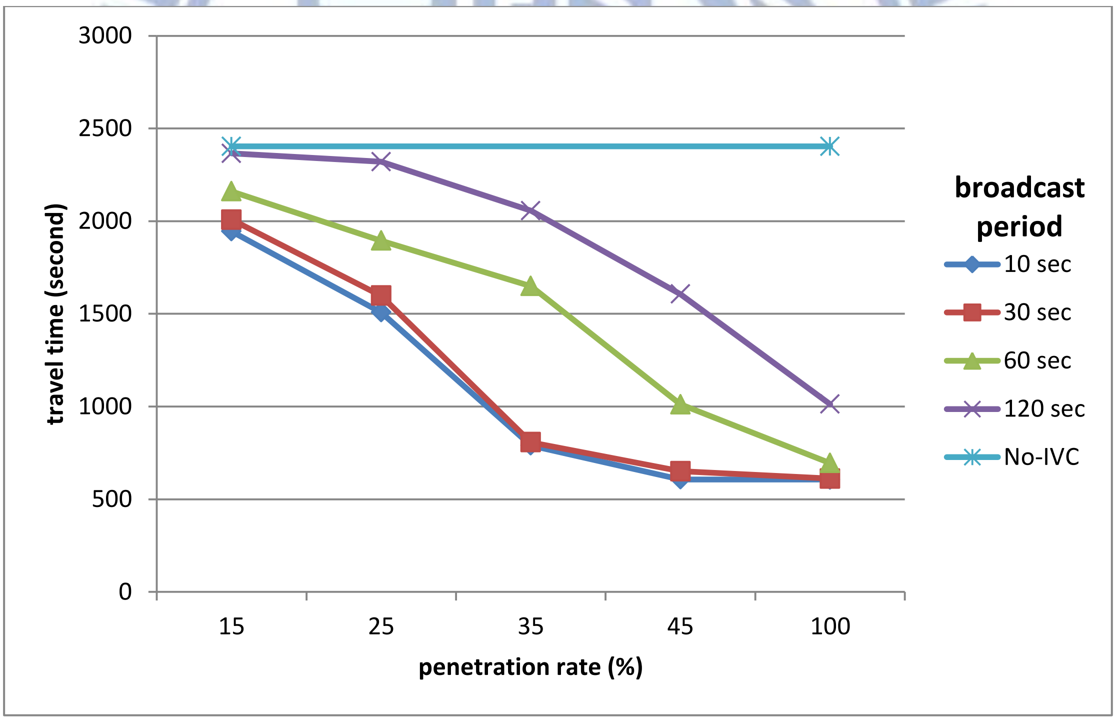
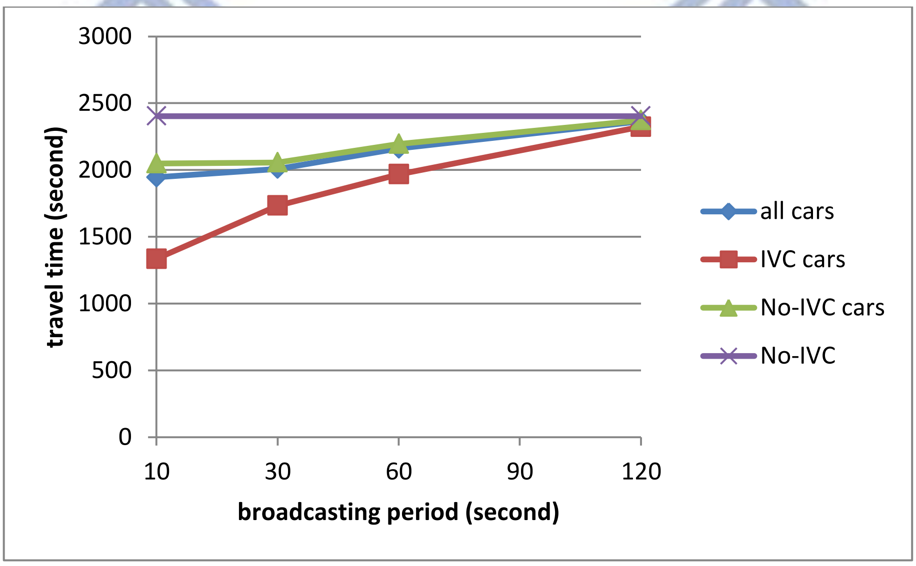
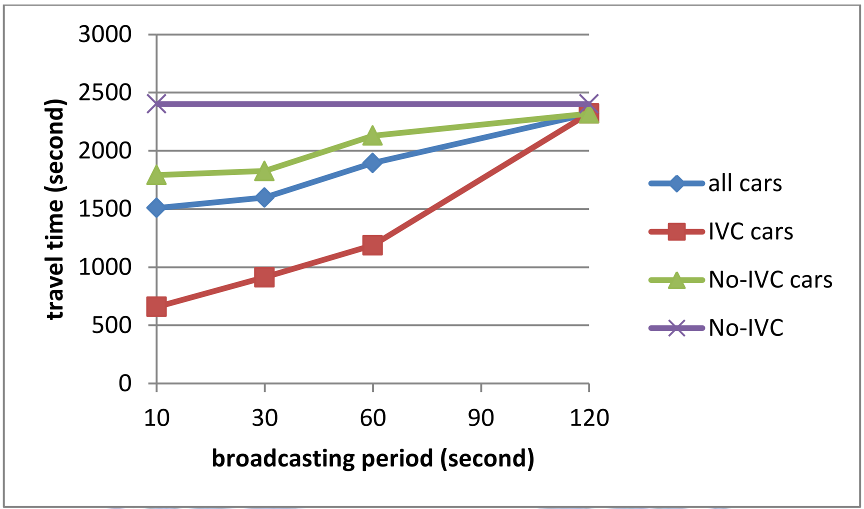


Fig. 4-4 The effects of penetration rate on the average travel time with different broadcasting periods

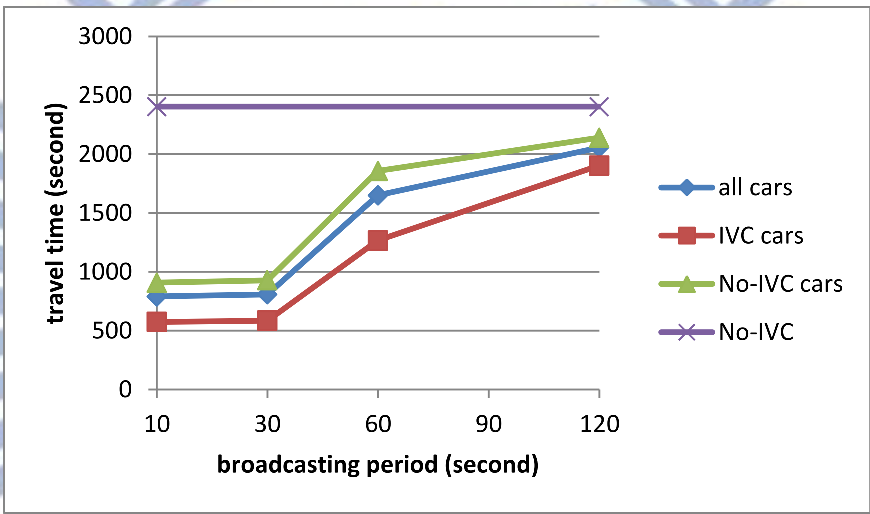
Fig. 4-5 shows the average travel time of the **navigation-equipped** cars and **navigation-unequipped** cars at four penetration rates. Blue line, red line and green line represents the average travel time of all cars, IVC cars (navigation-equipped cars) and No-IVC cars (navigation-unequipped cars), respectively. The x-axis is the broadcasting period, and the y-axis is the average travel time. Fig. 4-5 (a) depicts 15%, (b) 25%, (c) 35%, (d) 45% penetration rates of IVC cars. The results in Figs. 4-5 (a) and (b) indicate that when the broadcasting period is long (120 sec), the travel time of all cars cannot be reduced at low penetration rates (15% - 25%); the IVC cars and No-IVC cars have approximately the same travel time, but when broadcasting period is shorter, the IVC cars experience more significant time saving than No-IVC cars. Fig. 4-5 (c) shows that when penetration rate is 35%, the IVC cars have more significant time saving than No-IVC cars even if broadcasting interval is as high as 120sec. Finally, Figure 4-5 (d) depicts that highest penetration rate (45%) has the most time saving, and when the broadcasting period is short (10 - 60 sec), IVC cars and No-IVC cars have similar travel time. This phenomenon also shows that when the penetration rate is high (45%) and the broadcasting interval is short (10 - 60 sec), the navigation system has satisfactory traffic dispersing efficiency. In fact, a higher penetration rate of IVC cars enjoys more traffic dispersing efficiency.



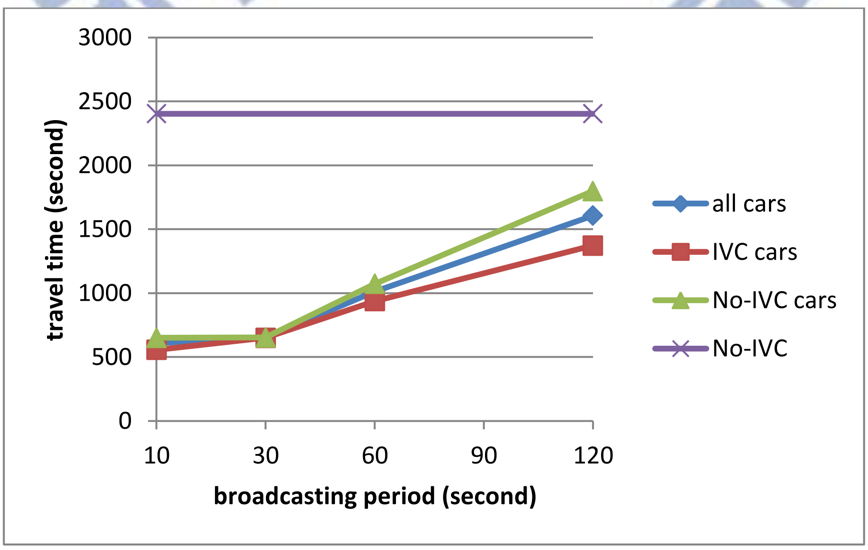
(a) The average travel time at 15% penetration rate



(b) The average travel time at 25% penetration rate



(c) The average travel time at 35% penetration rate



(d) The average travel time at 45% penetration rate

Fig. 4-5 Compare the IVC cars with No-IVC cars (with various broadcasting periods).

4.2.2 The Effects of Sending Buffer Size

The sized of sending buffer affects the data integrity. While the sending buffer is larger, more data samples can be propagated in each broadcast, and more complete traffic information may be obtained. Therefore we investigate the effects of sending buffer size with different penetration rates on the average travel time. The broadcasting period is fixed to 30 sec because 30 sec is short enough as we described, and the interval to perform rerouting is 30 sec.

Fig. 4-6 depicts the effects of sending buffer size on the average travel time. The multiple color lines represent the 0%, 15%, 25%, 35%, 45% and 100% penetration rates of IVC cars. The x-axis is sending buffer size and the y-axis is the average travel time. We can observe that the sending buffer size is negatively correlated to the average travel time. When the buffer size is larger than 250 bytes, average travel time decreases relatively slowly and 45% penetration rate has the average travel time approximating to that of 100% penetration rate. This is the reason why we choose 250 byte as our buffer size in previous experiments. We also notice when the buffer size is larger or equal than 1000 bytes, 35% penetration rate also has the average travel time approximating to 100% penetration rate. Note that 15% and 25% penetration rate can't an average travel time close to that of 100% penetration rate, no matter how large the buffer size is, and the lowest average travel time is about 1500 sec. This represents that we can lower the penetration rate (reduce the cost equipping IVC cars) within a limited range and obtain a similar average travel time as 100% penetration rate at a cost of using a larger buffer size. In addition, the average travel time of 45% penetration rate is slightly better than that of 100% penetration rate when the buffer size is larger than 1000 bytes. The reason is too many cars choose the same "best route” to cause secondary congestion when all drivers use CATE.

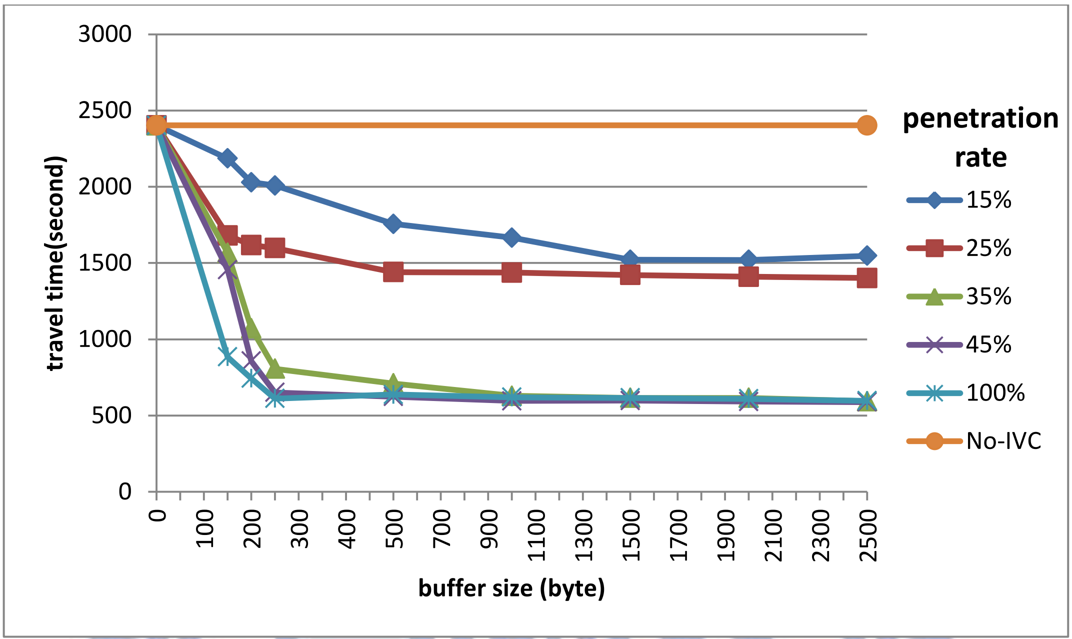


Fig. 4-6 The effects of sending buffer size on the average travel time

Fig. 4-7 depicts the effects of penetration rate on the average travel time with different sending buffer sizes. Each line represents a different sending buffer size: 150bytes, 200bytes, 250bytes, 500bytes, 1000bytes, 1500bytes, 2000bytes and 2500bytes. The x-axis is penetration and the y-axis is the average travel time. The results indicate that the average travel time decreases as the penetration rate increases for all sending buffer sizes. In each penetration rate, buffers of size 1500 - 2500 bytes have approximately the same travel time. This means 1500-bytes sending buffer is large enough for all each penetration rates. When the penetration rate is larger l than 25%, buffers of size 500 - 2500 bytes have approximately the same travel time, and when the penetration rate is larger than 45%, buffers of size 250 - 2500 bytes have approximately the same travel time. This phenomenon indicates more IVC users in the system can reduce the buffer size because a higher penetration rate leads to receiving more complete information.

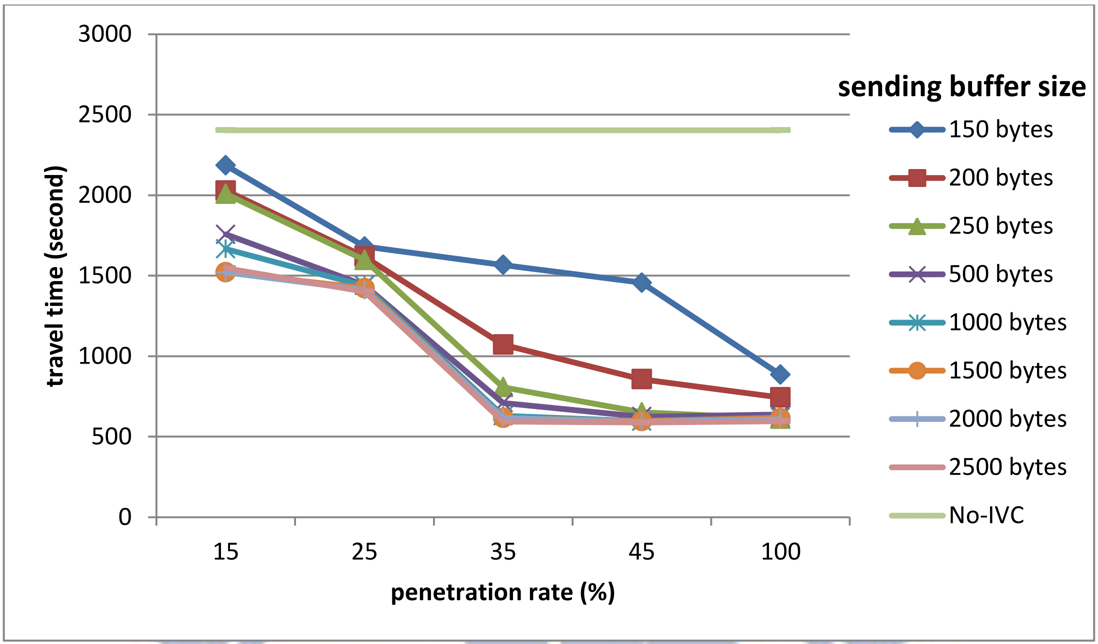
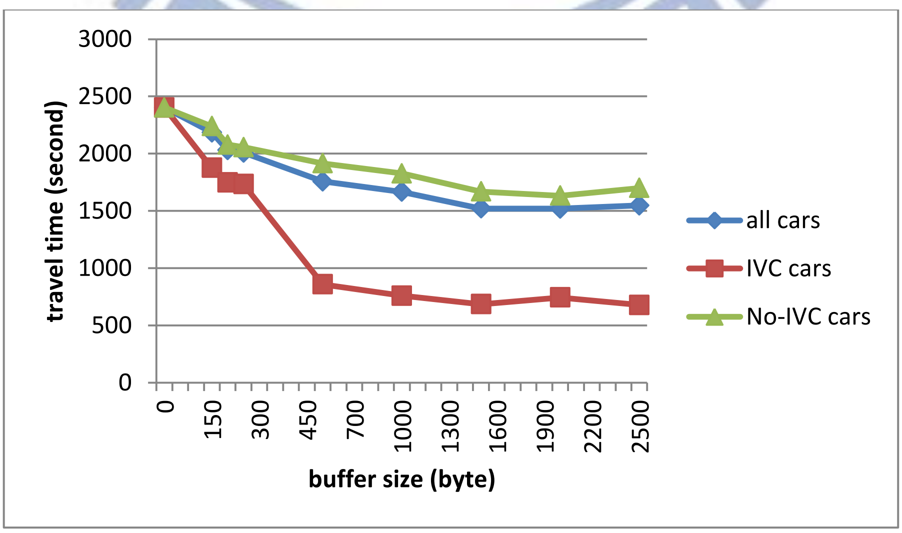
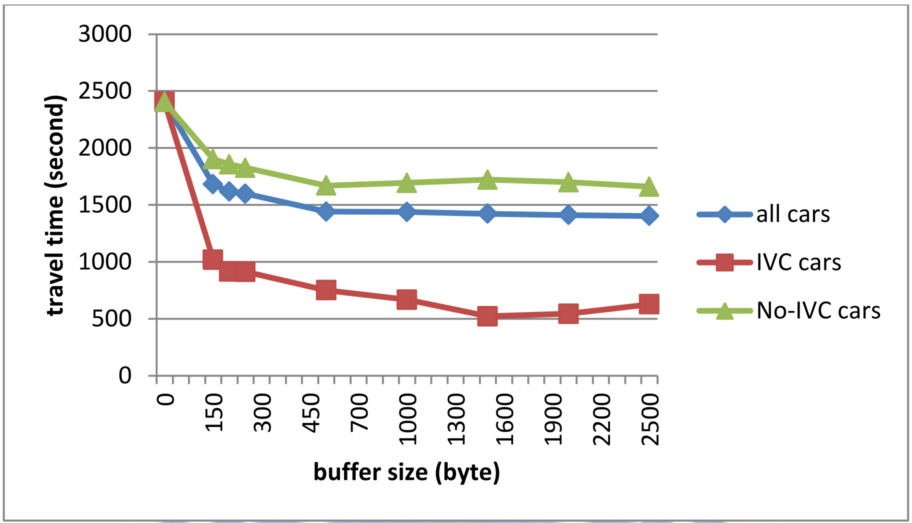


Fig. 4-7 The effects of penetration rate on the average travel time in different sending buffer sizes

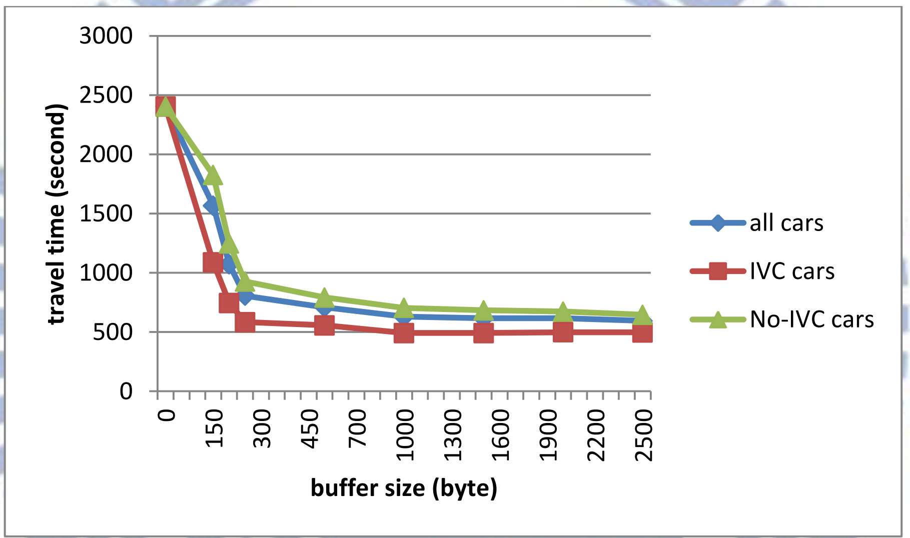
Fig. 4-8 compares the **equipped-system** cars with the **unequipped-system** cars at four penetration rates. The x-axis is the sending buffer size. When there are more IVC cars in the road network, the travel time difference between IVC cars and No-IVC cars is smaller. This phenomenon indicates that when the navigation system has a higher penetration rate and the broadcasting interval is short enough, it has more traffic dispersing efficiency.



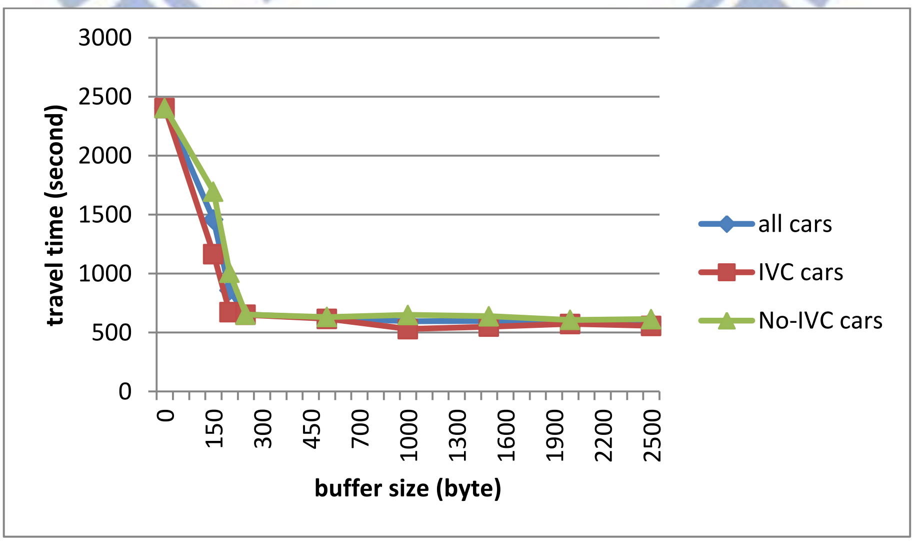
(a) The average travel time at 15% penetration rate



(b) The average travel time at 25% penetration rate



(c) The average travel time at 35% penetration rate



(d) The average travel time at 45% penetration rate

Fig. 4-8 Compare the IVC cars with No-IVC cars (with various sending buffer sizes).

4.2.3 The Effects of Left Turn Delays

We investigate the performance improvement of considering the left turn delay. The broadcasting period is 30 sec because 30 sec is short enough as we have described, and the interval to perform rerouting is 30 sec. 100% penetration rate of IVC cars is considered.

Fig. 4-9 depicts the average travel time when left turn delays un-considered (version 1) and considered (version 2) with different buffer sizes. Blue, red and green lines are version 1, version 2 and no navigation, respectively. We can observe that considering the left turn delay provides a shorter travel time when the buffer size larger than 500 bytes. To obtain the shorted travel time, version 1 requires a sending buffer of size at least 250 bytes, while version 2 requires 500 bytes, double the size of version 1. This can be expected because the data samples of version 2 are double (each road segment has TL and TN samples). When the buffer size is larger than 500 bytes, version 2 saves the average travel time by 17%, compared with version 1.

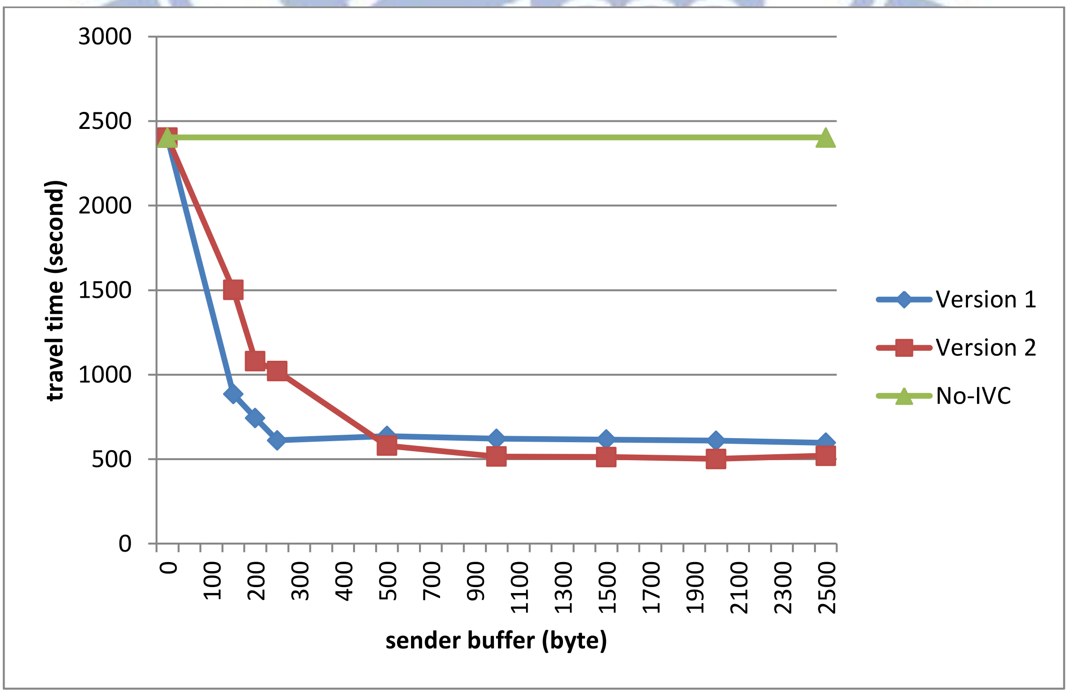


Fig. 4-9 The average travel time of version 1 and version 2

Furthermore, we investigate the detailed improvement of version 2 over version 1. The sending buffer size is fixed at 1000 bytes. The same as CATE [10], for cars with a shorter travel time using navigation, we consider the travel time decreasing ratio, a car's old travel time (without navigation) divided the new travel time (using navigation); for cars with a longer travel time, we consider the time increasing ratio, a car's new travel time / old travel time. Table 4-5 and Table 4-6 show the total statistics of travel time improvement in version 1 and version 2. As Table 4-5 showing, there are about 58% (3082/5328) cars with increasing travel time, 5% (267/5328) cars with the same travel time, and 37% (1979/5328) cars with increasing travel time. Similarly, In Table 4-6, we observe about 61% (3259/5328) cars with increasing travel time, 4% (220) cars with the same travel time, and 35% (1859) cars experiencing longer travel time. This indicates version2 have more 3% percentage of cars with decreasing travel time, and the percentage of cars experiencing longer travel time is also reduced by 2%.

Table 4-5 Total statistics of travel time improvement in version 1.

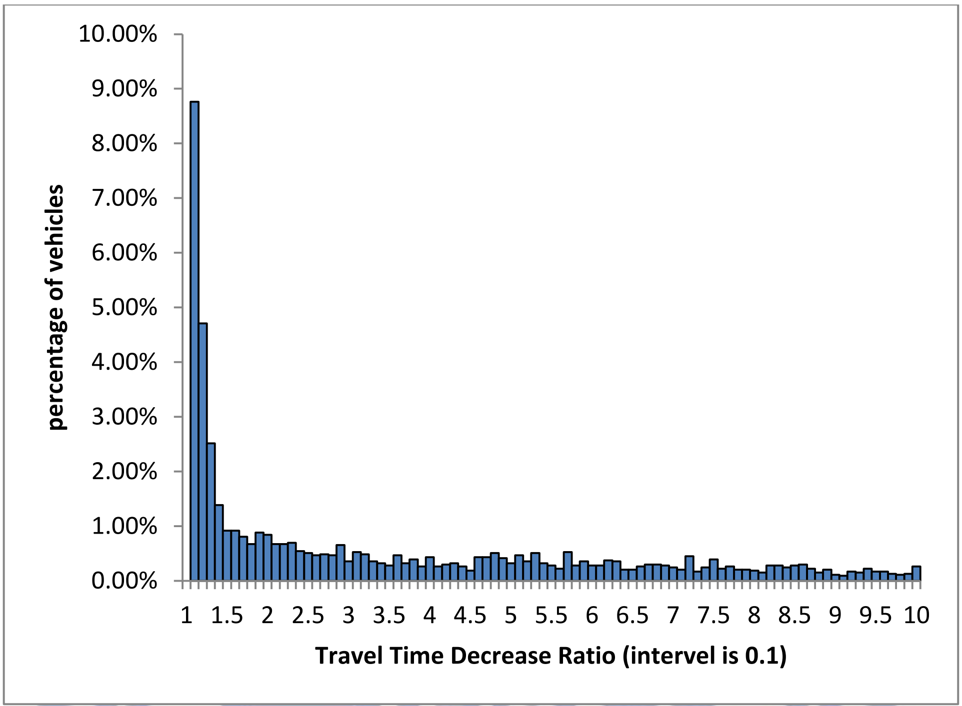
Total observed cars : 5328	With decreasing TT	With the same TT	With increasing TT
	3082	267	1979
	58%	5%	37%

Table 4-6 Total statistics of travel time improvement in version 2.

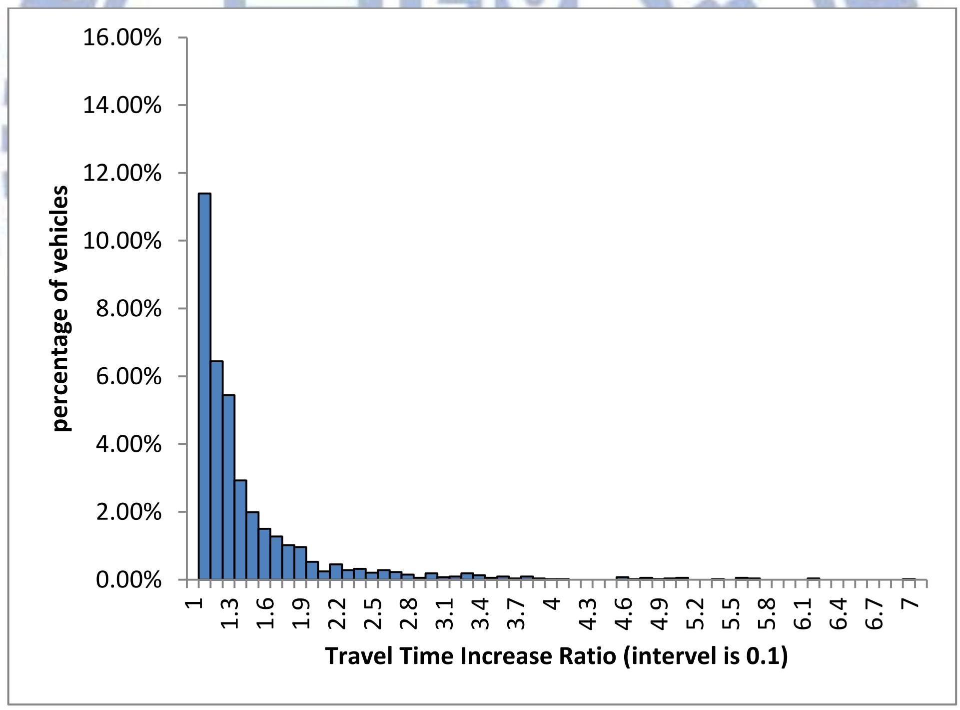
Total observed cars : 5328	With decreasing TT	With the same TT	With increasing TT
	3259	220	1859
	61%	4%	35%

Fig. 4-10 and Fig. 4-11 depict the distributions of the travel time decreasing ratio and increasing ratio for version 1 and version 2. The distributions are represented in histograms with an interval of length 0.1. The results in Fig. 4-10 (a) 22% (1194/5328) of the cars save less than 50% travel time (decreasing ratio1-2), and about 10%

(521/5328) percentage of cars have exceeding 10 times speed up (decreasing ratio larger than 10), but this is not showing at the figure. In figure Fig. 4-10 (b), we can observe lower increasing ratio has more percentage of vehicles. There are about 33% (1781/5328) vehicles experiencing less than 100% extra travel time, some cars increasing 2 - 4 times travel time , little cars increasing 5 - 7 (< 1%) times travel time and no car has larger than 7 increasing ratio. Similarly, the results in Fig. 4-11 (a) 23% (1233/5328) cars save less than 50% travel time (decreasing ratio1-2), and about 14% (728/5328) percentage of cars have exceeding 10 times speed up (decreasing ratio larger than 10) but this is not showing at the figure. In figure Fig. 4-11 (b), we observe about 34% (1804/5328) cars experience less than 100% extra travel time, little cars experiencing 2 - 5 times travel time (< 1%) and no car has larger than 5 increasing ratio. The results of Fig. 4-10 and Fig. 4-11 show that compared with version1, version2 have more 4% cars can get higher speed up (decreasing ratio larger than 10). It is worth mentioning that in version2, almost all cars with increasing travel time experience less than 100% extra travel time (1804/1859 =97%) and few cars waste more time (no car has larger than 5 increasing ratio). This is obviously better than version1 (1781/1979=90% cars increase less than 100% extra travel time and little car increase 5 - 7 times travel time)

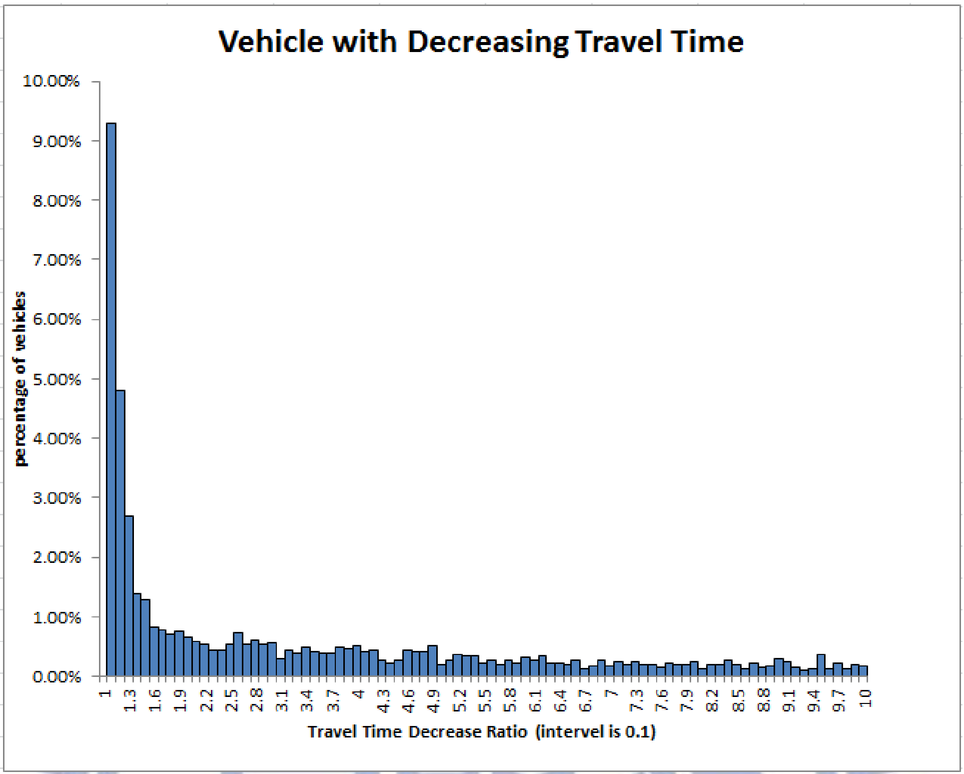


(a) The distribution of vehicles with Decreasing Travel Time

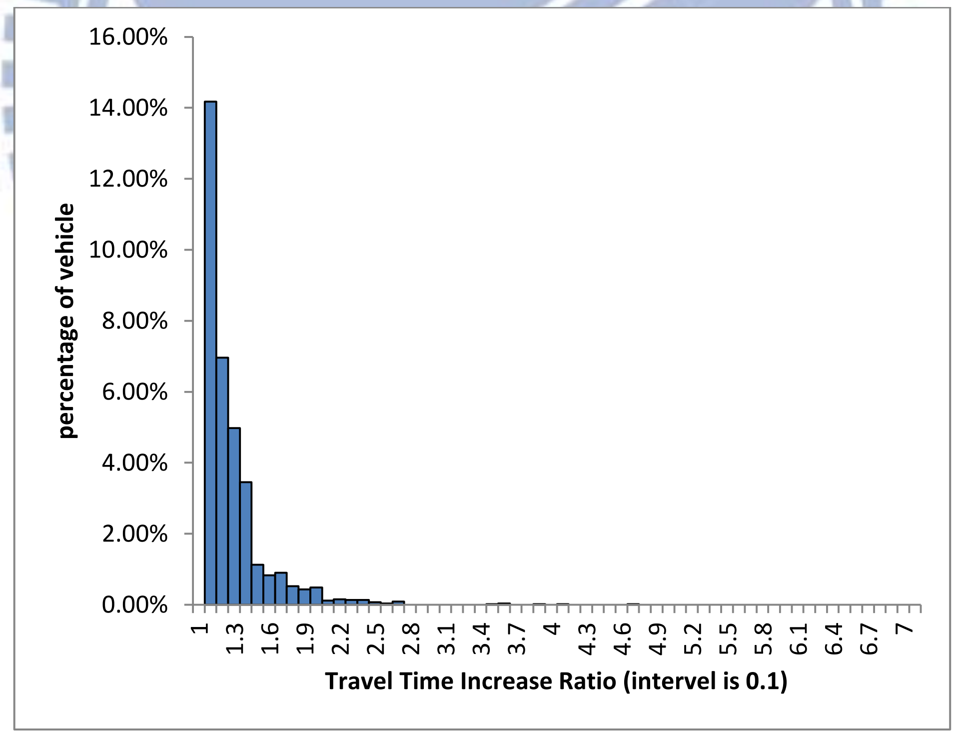


(b) The distribution of vehicles with Increasing Travel Time

Fig. 4-10 The distributions of travel time changing in version 1



(a) The distribution of vehicles with Decreasing Travel Time



(b) The distribution of vehicles with Increasing Travel Time

Fig. 4-11 The distributions of travel time changing in version 2.

Chapter 5 Conclusions

In this thesis, we implement and research a FCD-based vehicle navigation system that has a great scalability. This system consists of vehicles vehicle with a smartphone which is GPS-ability and vehicular ad hoc network (VANET) supporting. Each car shares the traffic information through the network interface, and uses this information to estimate current traffic condition for achieving the purpose of saving travel time. In order to further improve the performance of this navigation system (CATE), we also purpose a simple method to calculate the turning delay and effectively mitigate the intersection delay caused by turning delay under the limited bandwidth.

We simulate the system via Veins that that combines the network simulator OMNeT++ with traffic simulator SUMO to obtain the realistic performance evaluation. The simulation result of CATE (version1) indicates more buffer size and shorter broadcasting interval can improve journey efficiency. However, the improvement is limited (e.g. with a high penetration rate (45%-100%), 30-sec broadcasting period and 250-bytes sending buffer is enough in our experiment). Therefore, we purpose an extending version that provides the turning delay to further improve the performance, and the result indicate our version can further reduce about

17% average travel time, and the percentage of cars experiencing longer travel time is reduced by 2%.). We need the double buffer size to achieve this improvement. However, there are still about 35% vehicles using more time, and the road network is simple because limited resource. Therefore, the further researching and improvement is necessary.

In the future, we may try to predict the traffic condition and use a smart distribution technique of traffic information to optimize the performance of this system. A smart distribution technique can avoid that many cars receive the similar information causing them choosing the same “best route”. The accurate prediction not only can help driver to make a more effective route planning but also couple with a smart distribution technique of traffic information to avoid secondary congestion further effectively. A powerful navigation system can give drivers a comfortable journey.

References

[1] D. Middleton and R. Parker, “Vehicle Detector Evaluation”, Publication FHWA/TX-03 /2119-1. Texas Transportation Institute, Texas Department of Transportation, FHWA, 2002.

[2] P. T. Martin, Y. Feng, and X. Wang, “Detector technology evaluation”, Department of Civil Environmental Engineering, University of Utah-Traffic Lab, Salt Lake City, UT, 2003.

[3] TomTom, http://www.tomtom.com/

[4] IntelliOne, http://www.intellione.com/

[5] ITIS Holdings plc, http://www.itisholdings.com/

[6] Mediamobile, http://www.mediamobile.com/

[7] http://www.gps.gov/systems/gps/performance/accuracy/

[8] Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, “Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones”, 2008.

[9] Lars Wischhof, Andre Ebner, Hermann Rohling, Matthias Lott, and Rüdiger Halfmann, “SOTIS - A Self-Organizing Traffic Information System”, in Proceedings of the 57th IEEE Vehicular Technology Conference, 2003.

[10] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, M. Gerla, “On the Effectiveness of an Opportunistic Traffic Management System for Vehicular Networks,” IEEE Transactions on Intelligent Transportation Systems, vol. 12, n. 4, pp. 1537–1548, Dec 2011.

[11] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-based ad hoc routing,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, Jun. 2006.

[12] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve. Data Aggregation and Roadside Unit Placement for a VANET Traffic Information System. In VANET 2008: Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking, Sept. 2008.

[13] Joon Ahn , Yi Wang ;, Bo Yu , Fan Bai ;, Krishnamachari, B. “RISA: Distributed Road Information Sharing Architecture”, INFOCOM, Proceedings IEEE,pp1494-1502, 2012.

[14] Wegener, Axel, Hellbruck, Horst, Fischer, Stefan, Hendriks, Bjorn, Schmidt, Christiane and Fekete, Sandor P, "Designing a Decentralized Traffic Information System - AutoNomos", Proceedings of the 16th ITG/GI - Fachtagung Kommunikation in Verteilten Systemen (KiVS), 2009.

[15] A. Wegener, E. M. Schiller, H. Hellbruck, S. P. Fekete, and S. Fischer,“Hovering data clouds: A decentralized and self-organizing information system,” in Proceedings of International Workshop on Self-Organizing Systems, pp. 243–247, Sep 2006.

[16] A. Wegener, H. Hellbr¨uck, S. Fischer, et al. Autocast: An adaptive data dissemination protocol for traffic information systems. Proc. of the 66th IEEE Vehicular Technology Conference, 2007.

[17] Y.-C. Chu and N.-F. Huang, "An efficient traffic information forwarding solution for vehicle safety communications on highways," Intelligent Transportation Systems, IEEE Transactions on, vol. PP, no. 99, pp. 1-13, 2011.

[18] S.-L. Tsao and C.-M. Cheng, “Design and evaluation of a two-tier peer-to-peer

traffic information system," IEEE Communications Magazine, vol. 49, no. 5, pp. 165–172, May 2011.

[19] Kowen Lu, Chien-Ming Cheng, and Shiao-Li Tsao, "An Adaptive Routing Algorithm for Two-Tier Traffic Information System," The 9th Annual IEEE Consumer Communications and Networking Conference, pp. 311 – 315, 14-17 Jan. 2012

[20] Q. Yang, A. Lim, S. Li, J. Fang, and P. Agrawal, "ACAR: Adaptive connectivity aware routing protocol for vehicular ad hoc networks," Proc. International Conference on Computer Communications and Networks, pp. 535–540, Aug. 2008.

[21] C. Sommer, O. K. Tonguz, and F. Dressler, "Traffic information systems: Efficient message dissemination via adaptive beaconing," IEEE Communications Magazine, vol. 49, no. 5, pp. 173-179, May 2011.

[22] http://sumo.sourceforge.net/

[23] D. Krajzewicz, G. Hertkorn, C. R¨ossel, and P. Wagner, "SUMO (Simulation of Urban MObility); An open-source traffic simulation," in 4th Middle East Symposium on Simulation and Modelling (MESM 2002), Sharjah, United Arab Emirates, pp. 183–187, September 2002

[24] A. Wegener, M. Piorkowski, M. Raya, H. Hellbr¨uck, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators," in 11th Communications and Networking Simulation Symposium (CNS'08), Ottawa, Canada, April 2008.

[25] http://www.omnetpp.org/

[26] http://veins.car2x.org/

[27] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and

road traffic simulation for improved IVC analysis," IEEE Transactions On Mobile Computing, Vol. 10, No. 1, January 2011

[28] W.R. Reilly, C.C. Gardner, J.H. Kell A technique for measurement of delay at intersections Technical Report, Rep. No. FHWA-RD-76-135, Volume IFederal Highway Administration, Washington, DC (1976)

[29] S. Teply Accuracy of delay surveys at signalized intersections Transportation Research Record 1225Transportation Research Board, National Research Council, Washington, DC (1989)

[30] R.F. Benekohal, W. Zhao, Y. Lu, L. Wang Real-time delay measurement and intersection analysis system Proc. 4th Int. Conf. on Microcomputers in Transp.ASCE, pp. 285–296, New York (1992)

[31] F. Qiao, P. Yi, H. Yang, S. Devarakonda Fuzzy logic based intersection delay estimation Mathematical and Computer Modelling, pp. 1425–1434, 2002