THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

SHARING THE ROADS
USING ROUTE INFORMATION SHARING

by
Anthony Barber

A thesis submitted in partial fulfillment of

the requirements for the degree of Master of Science

in the Discipline of Electrical Engineering

The Graduate School

The University of Tulsa

2011

T H E   U N I V E R S I T Y   O F   T U L S A

THE GRADUATE SCHOOL

SHARING THE ROADS

USING ROUTE INFORMATION SHARING

by
Anthony Barber

A THESIS

APPROVED FOR THE DISCIPLINE OF

ELECTRICAL ENGINEERING

By Thesis Committee

______________________________________, Chair
Jinsong Zhang

____
Roger Mailler

____
Gerald Kane

COPYRIGHT STATEMENT

ABSTRACT

Anthony Barber (Master of Science in Electrical Engineering)

Sharing the Roads Using Route Information Sharing

Directed by Jinsong Zhang

75 pp., Chapter 7: Conclusions

(107 words)

Vehicular Ad Hoc Networks (VANETs) will provide useful services to drivers through communication equipment built-in to vehicles. Before the equipment for VANETs can be standardized, a more accurate assessment of the supported services must be made. The thesis in this research investigates different routing methods for use in VANETs. Shortest distance, live traffic map, and route information sharing methods are compared against a hybrid method proposed in this thesis. These are compared using microscopic simulations of the networks based on SUMO and several programs created for this research. Travel times, wait times, computation times, pollutants, fuel consumption, and bandwidth are used as criteria to compare these methods.

ACKNOWLEDGEMENTS

God has blessed me with a longer list of people than I can fit into this entire thesis. I am thankful to all of them, and I would like to highlight the ones most closely involved with this thesis.

Dr. Jinsong Zhang gave me the opportunity to return to TU and do this research for my Masters.

Dr. Roger Mailler and Dr. Gerald Kane served on my thesis defense committee and gave me quality feedback to improve this thesis.

The paper by Tomohisa Yamashita, Kiyoshi Izumi, Koichi Kurumatani, and Hideyuki Nakshima gave me the challenge that was worthy of a thesis.

The entire group at Project SUMO that developed the SUMO software and provide useful insight to questions via their mailing list, specifically Daniel Krajzewicz.

Patrick Lester wrote an excellent tutorial on the A Star algorithm which I repeatedly referred to while writing RouteMan.

Great teachers helped me gain wisdom and inspired a passion for finding out the answers to questions.

Lastly, I would like to thank the lady who takes out the trash in my lab in the late night/early morning hours. She reminded me that we are all working to make the world a better place, and that each of us has an important part in it, even if we do not seem directly involved in the big breakthroughs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Daily commutes regularly suffer from traffic jams caused by congested routes, wrecks, or construction Research in this thesis is working toward alleviating those pains. In it, I look at how vehicles can communicate to lower the affect of these factors on travel times.

1.1 Overview of the Problem

There are many problems associated with traffic congestion. Commuters are quick to observe that their travel times lengthen when they encounter congestion. For commercial drivers, this lost time in congestion means low efficiency. A driver's pocketbook notices the problems of congestion when they go to the gas station. During stop and go traffic, the regular acceleration and braking along with periods of stoppage while the motor runs all lead to decreased fuel economy [1]. Less noticeable is the increased wear on a car, causing increased maintenance costs. In 2008, the world's attention was turned towards another problem of congestion: decreased air quality for the Olympics in Beijing. In that example, one of the solutions to relieving pollution levels was to limit the number of cars on the road [2]. All of these factors contribute to another problem of traffic congestion: psychological stress. The modern problem of traffic has led to the phenomenon of road rage [3]. Though many drivers do not reach that level of trauma, we at least get a little frustrated when we encounter an unexpected delay. Traffic congestion causes many problems for drivers, but they are not the only shortcomings of our transportation system.

Traffic accidents, a common cause of congestion, are another problem on today's roadways. Cars can prevent accidents if they relay useful information to drivers [4]. Some accidents may be practically unavoidable. Therefore, we would like to have devices that ensure emergency personnel are notified in a timely manner. During the Oklahoma blizzard of February 2011, an SUV carrying 8 people drove off a bridge near Miami and fell 61 feet into the frozen Spring River [5]. Because another driver saw the accident and reported it, 5 of the passengers were rescued. Had no one been around, none of the passengers may have survived. Vehicles that automatically transmit distress signals could increase the survivability of this scenario [6].

A final problem worth noting is the ability to communicate from a moving vehicle. Recently, cell phone towers began dotting the American highway system and now allow people to get phone calls and internet data. Current communication with a car (or its passengers) with fixed infrastructure is a relatively expensive endeavor [7]. The construction, maintenance, and land purchase for a cell tower is not cheap. Another common option is satellite communication, which can be inhibited by forests, tall buildings and other objects that surround the road [8]. Traditional peer-to-peer computer networking methods do not work well when the computers are traveling away from each other in unforeseen routes. Even getting the relatively basic information of whether or not a car is at an intersection involves the costly tearing up of a road to place induction loops [9].

Whether vehicles are transmitting traffic updates, emergency messages, or user Internet data, there is a clear need for better communication methods on the roadways.

1.2 Overview of VANETs

Today's cars often come equipped with sensors that make them aware of their situations, as well as devices for communication. The long-term goal of research in this area is to tie these devices together with other transportation services to provide better services to drivers and transportation departments. Researchers are currently investigating Vehicular Ad Hoc Networks (VANETs) as the possible glue to bring these together [10] [11] [12] [9] [6]. VANETs use wireless communication between vehicles to relay data. The communication is not necessarily exclusive to vehicles, but can include entities such as traffic lights and government transportation centers (road side units). Because the topology of the network is constantly changing (as vehicles drive in and out of broadcast range of each other), new problems have to be investigated. In my first year as a graduate student at TU, I simulated the data exchanges of vehicles traveling on road networks. I followed the research of Tong Wu, who focused on the problems associated with vehicles broadcasting simultaneously [13].

At this point it is important to note that vehicle and communications terminology often overlap. Cars in VANETs both travel on a road "network" and transmit messages in data "network." Additionally, Tong's research examined emergency messages broadcast during an emergency situation, such as when two vehicles "collided." The emergency messages themselves could “collide” when two vehicles transmitted them at the same time, resulting in the loss of the message. To drive home the point, vehicles and messages both “travel,” from an origin to a destination (though electromagnetic waves propagating through air “travel” much faster than vehicles). When talking about VANETs, indefinite pronouns are confusing, so the language in this thesis is sometimes wordy in order to be specific.

The hardware for VANETs are expected to be devices hardwired into the vehicles. Each would contain a micro-computer, a wireless transceiver, and sensors (usually including a GPS). The long-term goal of the research here at The University of Tulsa is to build these devices and create a VANET among several cars and city traffic lights. Before the hardware can be selected, we need to know what computation and communication rates are required. In order to determine these requirements, we first have to decide which algorithms we plan on using. The research in this thesis examines a recently proposed routing algorithm, route information sharing, and investigates a method created here at TU that is a hybrid of the route information sharing and live traffic map methods.

1.3 Current Research

Standard vehicles can only use current data to determine their routes. This may include speeds of other vehicles on a roadway, construction schedules, or known points of heavy traffic. In [14], a new method is proposed that takes advantage of VANET data. The researchers introduced the route information sharing method. This method requires that vehicles broadcast their planned routes and adjust them based on the routes of other vehicles. The promise that route information sharing holds, is that vehicles can avoid areas that will become future bottlenecks, by knowing how many other vehicles plan on taking that route.

The researchers in [14] propose the method, and provide basic simulated evidence for its usefulness. That research is also presented in [15], however we know of no other articles that look into the route information sharing method.

1.3 Research Purpose

The purpose of the research presented in this thesis is to investigate how to use real-time traffic data to improve urban traffic. Specifically, I compare the usefulness of popular routing methods and evaluate the usefulness of the route information sharing method for use in VANETs, through computer simulation. I also propose a hybrid method and compare it with the pure route information sharing method. This research goes further than [15] because I use a microscopic simulator and realistic traffic patterns. The contributions made by this thesis are: the first microscopic simulations of the route information sharing method, the introduction of the hybrid model, a series of programs to simulate VANETs around the SUMO simulator, and an investigation of routing methods in terms of the criteria of travel times, wait times, computation times, pollutants, fuel consumption, and bandwidth.

CHAPTER 2

BACKGROUND

The following sections in this chapter provide background information on some of the important topics relating to this research. These do not provide an in-depth tutorial, but are meant to define some of the terminology used in this thesis (especially for phrases that might differ from paper to paper) and to give an understanding of the breadth of this research.

2.1 SUMO

To simulate the way that vehicles act on roadways, we chose the Simulation of Urban Mobility (SUMO) simulator. This simulator is open source, with most of the development coming from the employees of the Institute of Transportation systems at the German Aerospace Center [16]. Figure 1 shows SUMO’s visual user interface.

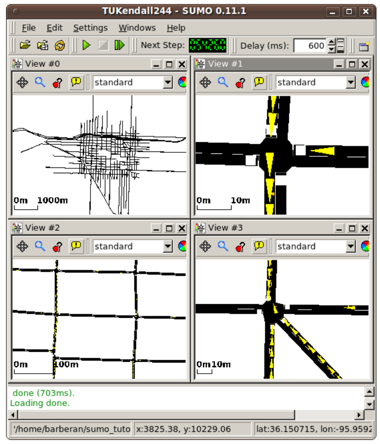

Figure 1 - SUMO GUI

SUMO is a microscopic simulator, which means that it operates at the level of each vehicle. Every vehicle is simulated. The vehicles react to other vehicles ahead of them, stop for traffic lights, wait for vehicles to pass before entering a roadway that has a higher priority, and follow their own individual route. SUMO can handle large simulation maps imported from reality, however we created our own scenarios for the research in this thesis. Initially, SUMO is given a listing of intersections, roads, traffic control lights, routes, and vehicles. Multiple types of vehicles can be simulated, but a basic car was chosen for the simulations in this thesis. The simulation can then proceed on its own. However I used SUMO's TraCI protocol to interact with it during the simulation, in order to gain more control over the simulation process. Using TraCI, SUMO runs for a specified period of time and then waits for further instructions. TraCI allows for vehicles, roads, intersections, and traffic lights to be controlled. Our simulation controller reads information about the vehicles, passes that information on to a routing program, and then passes the appropriate route changes back to SUMO through TraCI. Additionally, the maximum speeds of vehicles and road segments can be changed to simulate traffic accidents, construction, and other slowdowns. The research for this thesis was done using SUMO 0.11.0, though SUMO is up to 0.12.3 as of the time of this writing.

2.2 A Star

Simulations involving routing require an algorithm that will pick the "best" route based on the information available. Several routing schemes appropriate for vehicle routing exist. The research in this thesis uses the A Star, also known as A*, algorithm for picking the route. This algorithm is applicable to all of the routing criteria and methods discussed later. The idea behind it is simple, though it was perhaps the most problematic part of the coding for this research.

The A Star algorithm begins at the intersection where the vehicle will originate. That intersection, the originating point, is placed on the "closed" list. Each intersection that the vehicle is allowed to travel to from this intersection is then examined. The time it will take to get from this original intersection to the new intersection is calculated and stored as the new intersection's G score. The algorithm then estimates the time it will take to get from the new intersection to the destination, this is store as the new intersection's H, heuristic, score. The sum of these two scores is the F score for a new intersection. In a similar fashion, each of the other adjacent intersections is given an F score and placed on the "open" list with a pointer to the original intersection as their "parent" intersection. The open list is then looked at, and the intersection with the lowest F score on the open list is picked. The intersection is then placed on the closed list, and its neighbor intersections are examined. If they are not already on the open list, then they are placed on it with references back to this node as their parents. A G score is calculated by adding the previously mentioned G score to the time it will take to get to the new intersection. Therefore, it is now the total time it would take to get from the original node to the new intersection. If the node is already on the open list, then its previous G score is compared with the new G score. If the new one is lower, then the parent node is changed, and the H and F scores are recalculated for it. An example calculation can be seen in figure 2. This figure shows the calculation of the F score from the origin to point a and point d.

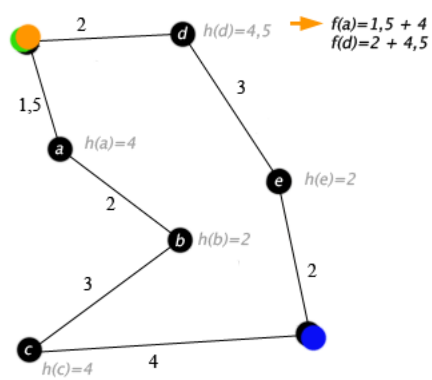

Figure 2 - Example of A Star Calculation [17]

At the beginning of each round, the node that has the lowest F score is chosen and its paths to its neighbors are inspected. When the destination node is reached, it has an H score of zero. It will eventually be chosen as the lowest F score, and placed on the closed list. When the destination node is on the closed list, the algorithm ends and the route can be determined by following the parent pointers back to the origin.

Estimating the H score of each node is important to the performance and reliability of the method. In order for the best route to be picked, the H score has to be the shortest possible time a vehicle could take to get from the intersection to the destination. This is heuristic admissibility, and it is critical to the optimality of the search. In the scenarios presented in this thesis, this is easy to determine because there is a maximum speed limit, and grid network requires that a vehicle travel vertically, then horizontally.

No matter which route the vehicle takes, the shortest distance will be the sum of the vertical and horizontal distance the vehicle will have to take to get to the destination. Using an H score value faster than the maximum speed limit will result in more intersections being examined, but will mean a wider search that requires more time. This is because the algorithm believes that there might be a road capable of that fast speed, and will search more nodes that are out of the way in hopes of finding the fast route. Using a smaller value may mean that the best route is not found and a poor route is selected. This is because the algorithm does not branch out far enough to routes that are out of the way, but end up being faster because of a faster traveling speed.

A similar algorithm, Dijkstra’s shortest path algorithm, is often also used for route planning. It is used by some of the SUMO tools. It is the same, except that it does not use an H score [18]. Therefore, the search spreads out more evenly, but ends up taking longer.

2.3 Greenshield’s Model

Getting a good approximation for the speed of vehicles is not exactly straightforward. One of methods used in preliminary simulations was simply averaging all of the vehicle speeds on a road segment and using that to approximate the speed on the segment. This seems like a good method, until traffic light control is included. If the only vehicle on a road segment is stopped at a stop light, this method will report the current average speed as 0 mph. Luckily, civil engineers have put more time into studying road speeds than I have. One of the common methods of approximating the speed of a road segment uses Greenshield's model [19]. In this method, there is a linear relationship between the estimated road speed (v) and the traffic density(k), as seen in the following equation:

$$v = vf \cdot (1-k/kj)$$

where kj is the jam density and vf is the free speed [19]. If there are no cars on the road segment, the density goes to zero and the estimated speed goes to the free flow speed. As the segment becomes more populated, the speed tends toward zero. The jam density is a tunable parameter in this model [19].

2.4 One-Time Routing Methods

Initially, all drivers have a path they plan on taking. Without updated traffic information for the road network, a vehicle will stay on the same path that it was initially routed with. Vehicles that do not receive updated traffic information are considered "one-time routing" vehicles. Their route is based on known information at the time of the initial routing. Two common criteria for judging routes are: which route is the shortest distance and which route will take the shortest amount of time.

2.4.1 Shortest Distance

In our simulations, the most basic routing method uses shortest distance as its criteria for judging routes. The goal of the method is to find the path from the vehicle's starting location to its destination that covers the least amount of distance. This is a straightforward method. The simulations in [14] included vehicles routed by this method.

Because the roads in that network all have the same speed limit, it is equivalent to a shortest time routing method, discussed next.

Shortest distance routing vehicles in this thesis are routed using the SUMO tool DUAROUTER. In fact, when vehicle routes are initially placed in SUMO, the shortest distance method is used for all vehicles, with the expectation that their proper algorithm will reroute them once they are on the road network.

2.4.2 Shortest Time

Travel time is the usual criteria drivers have for selecting their routes. The basic way to the judge the travel time of a route is to use the speed limits of each segment of a route. This is the way that posted speed vehicles work in our simulations. In addition, all of the "smart" methods in our simulations use travel time as their criterion for judging routes, and are rerouted using the A Star algorithm with a method for approximating the travel time on each road segment. Posted speed vehicles are one-time routing vehicles.

2.5 Live Traffic Map

With the advent of GPS, traffic cameras, other sensors, and various communication devices, drivers are aware as conditions on other road segments change. With new information, they may have reason to change their original route. Vehicles with a live traffic map can see current road conditions, such as average speed, number of vehicles or road hazards [6]. In our simulations, vehicles with live traffic maps have the current approximated speed on each road. In [14], these speeds are calculated using Greenshield's model. Although I tried other models in the preliminary research for this thesis, they did not perform better, so all of the simulations in this thesis will use Greenshield's model.

2.6 Route Information Sharing Method

Now that mobile communication devices are increasing in power, the ability of vehicles to share more information about traffic conditions has been expanded. The ability to look at live traffic maps with current data is useful, but route information sharing hopes to take traffic maps a step further, by projecting them into the future. The route information sharing method proposed in [14] operates on the assumption that if a driver know the routes that other vehicles plan on taking, they can avoid currently congested roads as well as roads that will become congested due to detouring vehicles. One scenario where this seems promising is when there is an accident on a major highway. There are a large number of vehicles on the highway, with more quickly queuing up behind them. If all of the vehicles exit at the last exit before the accident, the off-ramp will become a bottleneck, and some city streets will quickly become congested with as more vehicles begin to enter them. Theoretically, route information sharing will allow vehicles, or a centralized traffic control center, to compute new routes and cooperatively choose different routes so the added number of vehicles is dispersed throughout alternate routes.

The method proposed in [14] creates a "Prospective Traffic Volume" map for vehicles to use in comparing routes. This map is simply a listing of all the roads with an attributed value to account for expected congestion. In the paper, the value is referred to as the Prospective Traffic Volume, but it is not really a volume of cars. The Prospective

Traffic Volume is a product of the current feasible travel time (according to Greenshield’s model) and a value called the Total Passage Weight. The Total Passage Weight is the sum of each vehicle’s individual Passage Weight on that road segment. A vehicle calculates Passage Weights for each road segment in its route by counting the number of road segments from the destination to the specified road segment and dividing that by the total number of road segments in the vehicle’s remaining route. The calculated Prospective Traffic Volume for each value road segment can be used in a routing algorithm, such as A Star.

CHAPTER 3

ORIGINAL SOFTWARE

A significant contribution made by the research in this thesis was the creation of several programs to run VANET simulations in conjunction with the SUMO simulator.

3.1 Connection to SUMO through TraCI

SUMO does not have the route information sharing algorithm included in it. In order to implement the new routing algorithm, we needed to create a way to control the vehicles in SUMO during a simulation. This was accomplished via TraCI, an extension that provides a channel for communication with SUMO. With TraCI, SUMO acts as a server and listens for commands through a port. The library of commands in TraCI is extensive and includes control of each vehicle, traffic light, road, and almost every other variable in the simulation. When SUMO is called with the option to use TraCI, SUMO starts up, loads the scenario, and then waits for a command. Variables can be changed and then a command can be sent with how many seconds to run the simulation for before stopping and waiting for another command. For the simulations in this research, the actual connection is made through a python script, called Simulation Controller.

3.2 Overview of Simulation Software

The diagram in figure 3 shows the flow of programs and files for the simulations run in this research.

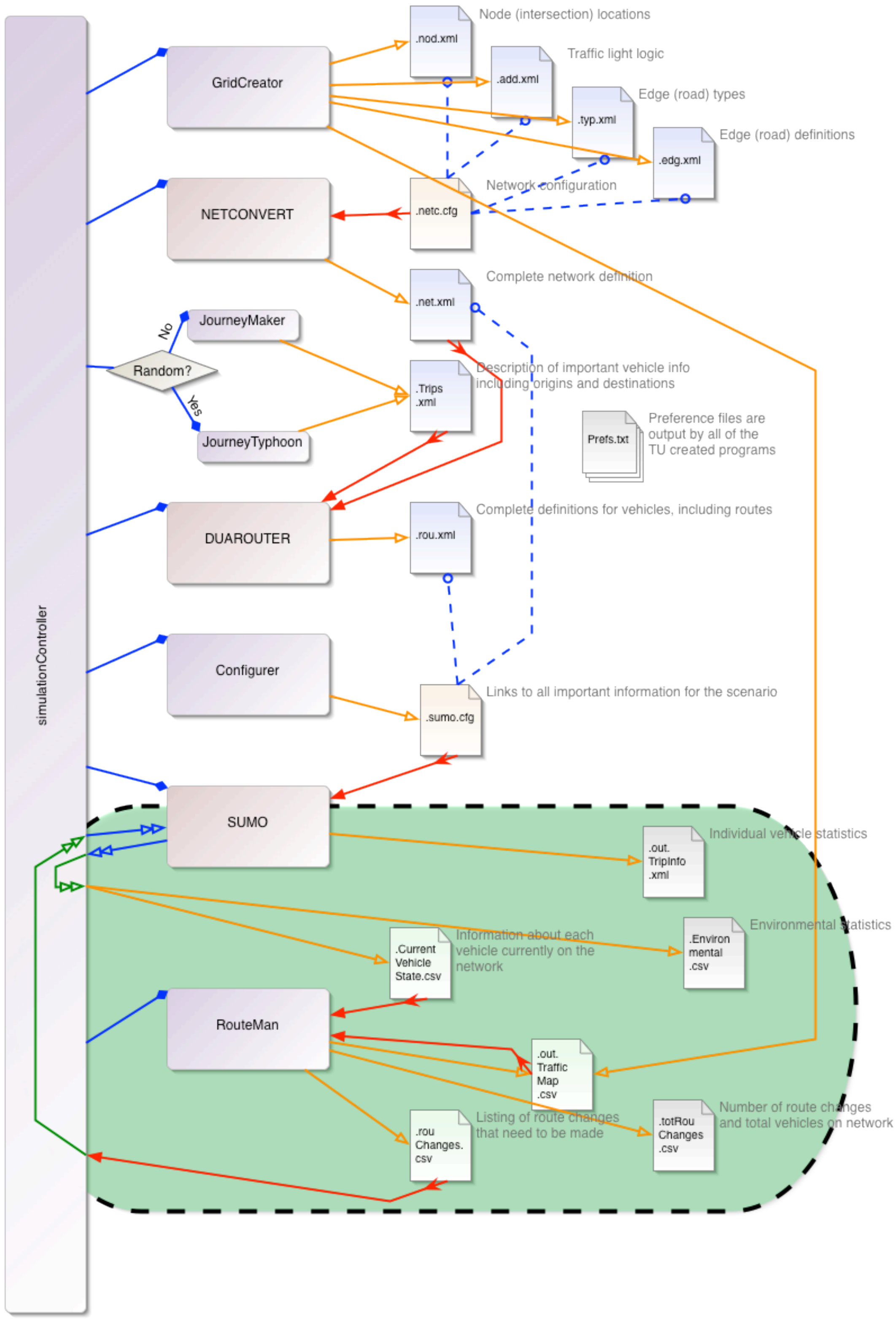

Figure 3 - The flow of a simulation

Each simulation is controlled by a Python script I created called Simulation Controller. This script calls each of the programs involved, starting with Grid Creator. Grid Creator is in charge of creating the road network used in the simulation. After the road network is defined, Simulation Controller activates NETCONVERT, a tool that comes with SUMO, which converts the road network files into a single file for use with SUMO. The next step is to define vehicles for the roads. Simulation Controllers choose either Journey Maker or Journey Typhoon, depending on the scenario, for defining these vehicles. The vehicle file created by either of the two journey programs is then fed into the SUMO's routing tool, DUAROUTER, which gives the vehicles their initial paths. The last preparation for the vehicles before they are fed into SUMO is to create a single file with links to the files that SUMO will read. This single file is created by Configurer. SUMO is started up in server mode by Simulation Controller, and then a loop begins for each second of the simulation with exchanges of gathering information about the vehicles, feeding them into my routing program, RouteMan, making the appropriate changes, and then running the next step of the simulation. The following sections go into more depth on each of the programs I have created for these simulation scenarios.

3.3 Simulation Controller

A single Python script, simulationController.py, handles the communication between the routing software I wrote and the SUMO software. This script contains the variables to be used in the simulation, the locations of the other programs, loops to handle gathering simulation information and special code for specific scenarios, such as how to simulate a traffic accident. Each program that Simulation Controller runs is opened in its own shell, allowed to run, and then the next task is handled when the shell connection closes. The only exception is SUMO, which has an open shell connection throughout the simulation phase.

3.4 Grid Creator

The road network in Tulsa is of major importance to us, because it is where TU hopes to place its first test VANET. The style of Tulsa's road network serves as a great model for our simulations because it is a very regular grid. In general, Tulsa has major city streets every mile, minor city streets every half-mile, and residential roads every tenth of a mile. Grid Creator, a program I wrote in C++, is meant to create this type of road network. Grid Creator sets up a rectangular grid, with the user choosing the intervals for major and minor roads, as well as the default type of road. Grid creator is capable of producing a uniform grid, like Tulsa's, or adding random lengths to roads to give it a rectangle shape. These are all passed as variables through the command line when it starts up. Grid creator outputs four files that define each intersection, the traffic light logic for those intersections, the types of streets in the road network, and the individual streets in the network. These files are xml files, conforming to standard SUMO tags. These files are for the SUMO tool DUAROUTER. It compiles them into a single file and expands on the definitions with detailed information such as specifics on connections within intersections and priorities for each lane.

3.5 Journey Creation

In the scenarios for this research, there are two philosophies for creating the origins and destinations for vehicles. One philosophy is that the places where vehicles are coming from and headed to are important; the alternative is that these places should be chosen at random. In [14], the road networks are filled using the later approach. This seems like a very scientific way to test the ability of their algorithm without having to look into differences such as areas of heavy traffic versus light traffic. The truth is that "This is a fast way to fill the simulation with life, but nothing that has something to do with reality" [20]. In reality, there are patterns to people's driving habits. In the morning, large numbers of people drive from residential districts and the suburbs to commercial and industrial districts. Around 8am and 3pm, traffic peaks around schools. At 5pm, there is a mass exodus from major cities to the suburbs. During the late evening, people are driving to and from residential areas and entertainment districts. Drivers cannot choose random starting points for their journey, they have to start where they last parked their cars. If drivers randomly picked their destinations, they would probably never make it to bed, they would be attempting to get home by process of elimination. If traffic were random, we would have little basis on where to place highways or major roads, because the traffic would be evenly distributed. However, I have noticed that fewer cars enter and exit Tulsa's road network via the street in front of my home, than the number of cars that enter and exit via the road in front of a major parking garage downtown. In [14], the roadways are all one-lane streets with uniform speed limits. Random routing is acceptable for a ballpark idea of how the route information sharing method performs. A random routing program, Journey Typhoon, is included for the sake of later discussion and comparison against the results of [14]. However, the philosophy of this research is that there are patterns to traffic, caused by the decisions people make, and the program for handling this kind of traffic is Journey Maker. Both are discussed in the following subsections.

3.5.1 Journey Maker

The purpose of Journey Maker is to write vehicle and journey definitions for specified origin-destination pairs. The user can specify the total number of vehicles they wish to create and the interval between the departures of the vehicles. Additionally users specify the probability weighting for selecting a vehicle routing type and the probability weighting for selecting an origin-destination pair. Journey Maker outputs an xml file containing definitions of: each vehicle type, the road they should originate on and the road that is their destination. This xml file has an extension ".trips.xml," conforms to a SUMO standard, and will be used in the SUMO tool DUAROUTER, which will use the network file to create initial routes based on the shortest distance criterion.

3.5.2 Journey Typhoon

A random routing program is also used in this research for specific scenarios that look at the ability of the routing methods to handle constantly changing traffic patterns. Journey Typhoon is the program used for generating the random origin-destination pairs in these scenarios. Journey Typhoon is unique in that it can create semi-realistic traffic patterns through random streams of vehicles converging at a popular destination. The terminology in Journey Typhoon will be explained through the following example. A specific scenario has a number of “typhoons” defined by the user. Each typhoon is a random destination intersection. There are a number of “streams,” which are random origin intersections that all go to that typhoon destination. A number of vehicles will start from each stream’s origin intersection. In the code, a typhoon destination is selected, then the random starting points of each stream are created and the vehicles originating from that stream are written. In the scenario, a typhoon occurs with a series of vehicles being created at random starting points but all headed toward the same destination. After the vehicles for that typhoon have entered the network, another set of vehicles is created for a new destination. The origin of the name for this scenario is the image created by this: lines of vehicles converging on the “eye” of the typhoon from locations around it. The user specifies the number of vehicles in each stream, the time interval between the start of each vehicle, the number of streams per typhoon and the number of consecutive typhoons to create. Although the typhoon pattern is available, the method actually allows for a variety of scenarios. A completely random scenario, like in [14], can be generated by assigning one vehicle per stream and one stream per typhoon. This way a vehicle is created with a random origin-destination pair, and then another vehicle is created with a new random origin-destination pair. Generating vehicles on a single route can also be accomplished, but by specifying one typhoon and one stream. As with Journey Maker, the actual routing will be handled later by the SUMO tool DUAROUTER. Journey Typhoon simply outputs the vehicle definitions with their origin-destination pairs in an xml file also has the extension “.trips.xml” and conforms to a SUMO standard.

3.6 Configurer

Configurer is a simple program that writes a configuration file for SUMO. The configuration file includes the port number that TraCI will communicate on and links to the vehicle routes file and the grid network file.

3.7 RouteMan

Routing decisions during the simulation are handled by RouteMan. This is the most complicated program built for this research, though it has been simplified over many iterations. Not only does it make the routing decisions for vehicles, it also does much of the statistics gathering. I feel I have commented the code extensively, so here I will only go over the main flow of the program. The code is available from me upon request. It is worth noting that RouteMan has been built so that it is simulator independent. As long as it receives a list of current vehicle information, it will output the list of vehicles that need to change their routes and those changes. Therefore it can work with any simulator or even a real VANET. Also it can simply be modified to output the traffic map and route information sharing map files for broadcast. This feature was disabled because it added a significant amount of time to the simulation and was not necessary.

After initializing preferences and necessary data, RouteMan determines the updates that will need to be made. The user can define how often the basic traffic map is updated (which relates to how often vehicles would broadcast their speed and positions), how often the traffic map is broadcast to vehicles (for the simulations, this means recalculating the live traffic map cars), and how often route information sharing broadcasts exchanges occur.

It should be noted that in the simulations a route information exchange takes place instantaneously between steps. Each step in SUMO is one second long. Therefore, these exchanges can be thought of as taking one second, but everyone reacts at the end of the second. Later in this thesis, there is a discussion on the required data rates for this to occur. As in real life, vehicles in SUMO take time to readjust for route changes. For example, if a vehicle needs to get over to another lane for an upcoming turn, it will have to find a window to merge. Therefore, although transmission occurs instantly in the simulation, the effects take enough time that the impact of this issue might be lessened. This uses the assumption that vehicles are making their own routing decisions, an alternate approach is that the servers updating the traffic maps can determine the routes for vehicles. Part of the research presented in this thesis is the effect of the follow-up broadcasts. Follow-up broadcasts are meant to give vehicles the opportunity to avoid picking the same alternatives by repeating the route information sharing process through several iterations in hopes of finding an "ideal" solution. In large complicated networks there might not be a solution that approaches ideal, but in smaller networks it might be more efficient to do the iterations on the servers and then send the updated routes to the vehicles in a single broadcast at the end. The reason for follow-up broadcasts as opposed to constant updates is it has been suggested that the communications channel may be shared by other services and therefore the time available for route information sharing may be limited to a portion of the time.

RouteMan determines if any vehicle type needs to update. If any do, it imports the list of vehicles. Vehicles that do not broadcast any information are not imported. Based on the vehicles, RouteMan decides which maps need to be updated and what information about the network needs to be updated. After all the information has been imported, the routing process is run on any vehicle that requires it.

The A Star algorithm is used to reroute vehicles. The heuristic score is determined based on an option passed by the user, and should be the quickest that any vehicle can travel. It assumes that a vehicle must move vertically then horizontally to reach its destination. This is a requirement of a rectangular grid network, but it can be easily modified to instead calculate an “as the crow flies” score that is useful in any network shape. The G score addition for each road segment is calculated based on the vehicle routing type. Each routing method has a single function that calculates its score. Vehicles that use hybrid routing are initially rerouted using the live traffic map method.

Before writing the output files, RouteMan checks to see which updated routes actually changed from their previous routes. If the route did not change, it will not be written in the output file. If a hybrid route did change, the percentage of common road segments between the old and new routes are calculated and checked against the hybrid threshold. At this point the hybrid vehicle is rerouted using the route information sharing method if it needs to.

Finally, the output files are written and RouteMan ends. The route change file will be used by TraCI (via Simulation Controller) to make the appropriate changes in SUMO. Additional output files have been added at times to RouteMan for statistics purposes, such as gathering overall average speed of vehicles in the Accident scenarios. The source code of the version of RouteMan used for each simulation in this thesis was manually archived with those simulations.

CHAPTER 4

SIMULATION SCENARIOS

The simulations used in this research are microscopic simulations. The maps are not based on a real location, but follow the style of the road network in Tulsa, Oklahoma.

4.1 General Settings

All of the results in this thesis are generated by running each scenario with only one routing method at a time. Additionally, the vehicle departure times and origin-destination pairs were kept the same in each simulation. However, a vehicle may be delayed from entering the road network if the roadway is already crowded. Therefore, it can be imagined that each of the vehicles has a scheduled time for turning on their engine and attempting to leave, though the vehicle may not be able to immediately get out of its parking lot if there is too much traffic on the roadway it wishes to enter.

4.1.1 Road Types

Several road types were created for our initial simulations, but only three road types are used in the simulations presented in this research. These road types are meant to replicate common ones seen in Tulsa, OK. They are a residential road, minor city street, and major city street. The characteristics of each are summarized by the table below.

Type	Priority	Lanes	Speed Limit (m/s)	Speed Limit (mph)
Residential	2	1	11.17	25
Minor	3	2	13.41	30
Major	4	2	20.11	45

Table 1 - Road types

All the scenarios presented in this thesis have a major street at every tenth street, a minor street every fifth street, and residential roads for all others.

4.1.2 Traffic Light Logic

The traffic lights in the simulations presented for research in this thesis are all static traffic lights with fixed cycles. Dynamic traffic lights based on induction loops can be simulated using TraCI coupled with vehicle sensors in SUMO, but this requires significantly more computing power for large networks and may be obsolete if VANETs are implemented with the ability to affect traffic lights. The fixed cycle nature of the traffic lights used in this research is notable because this will cause a difference between the results from the simulations in this thesis, and the results seen in previous route information sharing simulations. The simulations in [14] did not include any sort of traffic control, and the effects of this difference will be discussed later in thesis. Traffic lights are simulated at intersections of city streets. Vehicles on residential roads must yield to traffic on minor and major streets and find a large enough gap in traffic to safely enter or cross, including the time it takes to accelerate into the roadway. The timing for the cycle is listed below.

Street Type	Green Arrow	Yellow Arrow	Green Light	Yellow Light
Major	10	3	20	3
Minor	5	3	10	3

Table 2 - Traffic light phases

Intersections of a major street with a minor street use the same timing. The lights facing the major streets use the major street timing, and the lights facing the minor streets use the minor street timing. This is the same on edge roads. Corners have lights that are always green, because traffic does not cross lanes.

4.1.3 Stop Signs

SUMO does not contain a built-in stop sign feature, therefore stop signs had to be simulated using traffic lights. The traffic light cycle for simulating a stop sign gives one way of traffic a green light for 1 second, then all intersections have red lights for two seconds. The allowed traffic changes in the counter-clockwise direction so that the next car allowed to pass is the car on the right side. This simulates the behavior that all-way stops operate in the United States.

4.1.4 Vehicle Characteristics

Each routing type is defined as its own vehicle type in the SUMO files, however the characteristics of each vehicle are the same. The vehicle characteristics are displayed below.

Acceleration	Deceleration	Length	Max Speed	Driver Imperfection
3 m/s^2	6 m/s^2	5 m	50 m/s	0.5

Table 3 - Vehicle characteristics

Multiple vehicle types are declared for the sake of having a different id tags that can be used to easily determine the routing that needs to be applied.

4.1.5 Notable Differences from Previous Research

The simulations in [14] use single-lane roads with uniform speed limits, and no traffic control. The authors state that they chose “a simple traffic flow model to examine the interdependence between traffic congestion as macro phenomena and route choice of individual drivers as micro behavior” [14]. It is unknown if vehicles took into account the affects of vehicle acceleration and following distance, but this would seem to indicate that they did not.

Another difference is that [14] rerouted vehicles and exchanged route information every time a vehicle got to an intersection. Vehicles in real life should not decide to change the path they want to take when they are already at an intersection; this would cause congestion as vehicles had to cross over lanes of traffic to get into or out of a turning lane. The default in this thesis is for broadcasts to occur every second. Later, there will be a discussion on the results of changing this broadcast rate.

As far as I am aware, this thesis is the first research of route information sharing using a microscopic simulation for vehicle behavior and congestion.

4.2 Scenarios

The following scenarios are used for the simulations presented in this thesis. They were chosen because they highlight the differences in the behavior of route information sharing methods compared to the traditional method live traffic map method.

4.2.1 Accident

The main scenario used for simulation is the Accident scenario. This is a simulation of a four square mile grid network, 21 roads by 21 roads, with three streams of constant traffic. Every two seconds a vehicle enters the roadway, until a total of 5,000 vehicles are created. At 5,000 seconds, halfway through the creation of vehicles, an accident is simulated by changing the speed limit to zero on the roads leading to a main intersection. The accident lasts for 30 minutes before the intersection is reopened. This scenario is particularly useful because it contains two types of traffic patterns: the constant stream of vehicles in the beginning and end, and the congestion created by rerouting vehicles during the middle. The diagram in figure 4 summarizes this scenario.

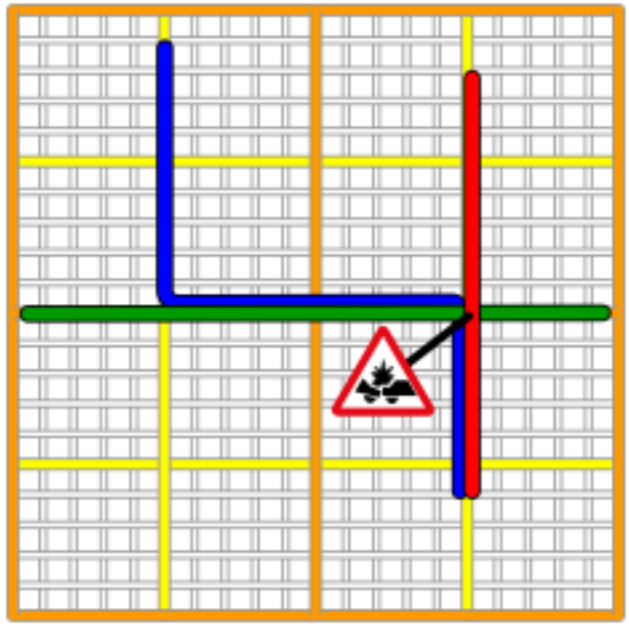

Figure 4 - Accident scenario

Although the vehicles do not have to pass through the stopped intersection, the intersection would be part of the “ideal” route for each origin-destination pair. Even if the routing methods do not use the intersection for all of the traffic, the accident will cause new congestion due to cars diverting from that intersection.

4.2.2 Random

Completely random traffic is unrealistic. The reasons were previous discussed at length in the thesis. Nevertheless, a completely random scenario was done in [14], therefore it is included here as a means to compare these results with the previous results. The grid for the Random scenario is an elongated network as seen in figure 5.

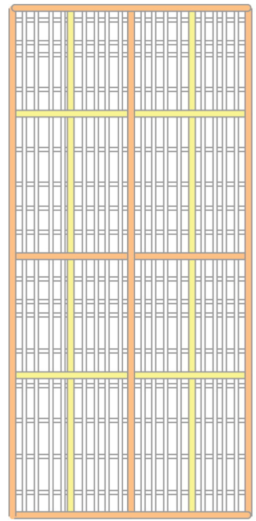

Figure 5 - Random scenario road network

The shape of the map has two useful characteristics. The first is that it allows for longer routes so that there is a more significant difference in the times of each routing method. The second is to allow a challenge for the methods to weigh the sacrifice of going out of their way against the savings of traveling a long distance over a road at a faster speed limit. Vehicle origin-destination pairs are generated randomly, and a new vehicle is generated every second up to 2,500 vehicles.

4.2.3 Hurricane

The purpose of the Hurricane scenario is to see the results of a scenario where congestion is unavoidable and unpredictable. The Accident scenario has constant congestion. The Random scenario gets congestion evenly distributed. The Hurricane scenario is unique because streams of vehicles are sent to the same location. This scenario uses the Journey Typhoon program to generate a traffic pattern with settings of 100 typhoons, 10 streams per typhoon, and 5 cars per stream. This results in 5,000 cars.

There are two seconds between each car in the stream entering the network. The map for the Hurricane scenario is the same as the one used in the Random scenario.

CHAPTER 5

PERFORMANCE COMPARISON OF ROUTING METHODS

The performance will be measured on four routing methods: shortest distance, live traffic map, and hybrid. The first three were examined in [14]. The fourth is a new method being introduced in this thesis. These routing methods will be compared primarily based on their vehicle travel times. This thesis will also investigate the amount of time vehicles are stopped, bandwidth used, processing time, environmental pollutants produced, and fuel consumed by each routing method. Each simulation will use only one routing method at a time. The effect of mixing vehicle types was examined in [14], but will not be examined in this thesis.

5.1 Shortest Distance Method

The shortest distance method is the simplest routing method. This routing method uses only static traffic map data. Therefore, vehicles using this routing method do not exchange information with each other. Shortest distance vehicles are routed one time, based only on the knowledge of the road segment lengths from a static traffic map and are not rerouted. In our simulations, shortest distance vehicles are routed using the DUAROUTER tool included with SUMO.

5.2 Live Traffic Map Method

Vehicles that route based on the live traffic map method have the basic ability to communicate with other vehicles. These vehicles have the ability to send their positions and receive updated traffic maps. Whenever an update to the traffic map is to be made in our simulations, RouteMan imports the road segments that each live traffic map vehicle is on. RouteMan uses this to calculate the feasible travel time based on Greenshield's method. As A Star creates the new route for live traffic map vehicles, it uses the feasible travel time as the road segment cost. The traffic map used to route live traffic map vehicles is calculated using all "smart" vehicles, which are any vehicles capable of broadcasting their positions. However, live traffic map vehicles do not broadcast their routes and have no knowledge of the routes of other vehicles.

5.3 Route Information Sharing Method

Route information sharing vehicles have the ability to send and receive not only position information, but also route information. The route information sharing vehicles in our simulation operate similar to the way that they were introduced in [14]. The Prospective Traffic Volume is still calculated in the same manner (discussed in background). However, the route information sharing vehicles in the simulations are routed only at broadcast intervals (the default broadcast interval is 1 second). This is not the same manner as in [14], in which the vehicles are routed whenever they reach an intersection. This is a necessary change in a simulation with multiple lanes, because (as discussed earlier) a vehicle changing its route at an intersection may have to merge across lanes and would hold up traffic.

5.4 Hybrid Method

A new type of routing method, which I will term the hybrid method is proposed in this thesis. The hybrid routing method is an attempt to get the benefits of both the live traffic map method and the route information sharing method. The hybrid method defaults to the live traffic map method but switches over to the route information sharing method if there is significant change in the vehicle's route when traffic information is updated. A significant change in a route is defined as having a percentage of common segments between the old and new route below a threshold value. The pseudo code for this method can be written as:

```
if (there is a new traffic map)
    reroute vehicle using live traffic map method;
    remove any road segments from the old route that have already been passed;
    count the road segments that the new route and old route have in common;
    divide the common number of segments by the longer of the two routes;
    if (common segment percentage < threshold)
        reroute vehicle using route information sharing method;
    send new route to vehicle;
end;
```

Our default threshold value for the hybrid routing method in this paper is 70%. The affect of the threshold value will be investigated later.

5.5 Performance Indices Used

The primary indicator of how well a routing method performs is the travel time of a vehicle. In the simulations, SUMO will track the travel times. When a vehicle is scheduled for placement on the network, SUMO attempts to put it on at the specified originating point. When the vehicle enters the roadway, SUMO begins a duration clock for that vehicle. If there is not enough of a gap in traffic to safely do so, SUMO starts a depart delay clock and waits to try again until the next step. The depart delay clock stops whenever the vehicle is finally placed on the road network. The duration clock stops whenever the vehicle reaches its destination. The travel time used to compare routing methods will be the sum of the duration and any depart delay. The reason for including the depart delay is that congestion preventing the vehicle from entering the roadway is unwanted. In real life, a delay that keeps a driver from getting out of their parking lot at work is an inconvenience and takes away time that they could be doing other things.

A comparison of the vehicles will also be made based on the amount of time that vehicles are stopped. Any time that a vehicle goes below 0.1 m/s, SUMO starts the waiting timer for that vehicle [21]. It stops any time the vehicle goes back above 0.1m/s, but it is a cumulative clock across the simulation. Stopping a vehicle causes wear on the brakes and wastes fuel while idling.

A very important factor in hardware selection for VANETs is the amount of bandwidth that the VANET will need in order to reliably operate its services. Therefore, the amount of bandwidth used at different broadcast rates will also be examined. Another factor in hardware selection for VANETs is the processing power required. A quick comparison of CPU usage will be made using the real time each smart method takes to run a simulation.

There is a lot of interest in the environmental impact of road traffic. SUMO allows users to track environmental statistics according to the Handbook of Emission Factors for Road Transport (HBEFA) standard [22]. The emissions are not a simple function of the amount of time a vehicle is on, therefore the different routing methods will produce different results based on their preference for things like driving over wait, etc. By the same method, SUMO also tracks fuel consumption. This data will be used to do a fuel economy comparison of the routing methods.

CHAPTER 6

RESULTS

Each routing method was run individually through each of the scenarios. The results are compared based on the criteria of travel times, wait times, computation times, pollutants, fuel consumption, and bandwidth.

6.1 Average Travel Times

When I want to get somewhere, I usually want to get there in the shortest amount of time possible. I do not think I am alone. Therefore, one of the most interesting statistics I am interested in looking at is the average travel times for vehicles using the different routing methods.

6.1.1 Accident Scenario

A realistic portrayal of traffic patterns is exhibited in the Accident scenario. The overall average travel times for vehicles using each routing method are seen in figure 6.

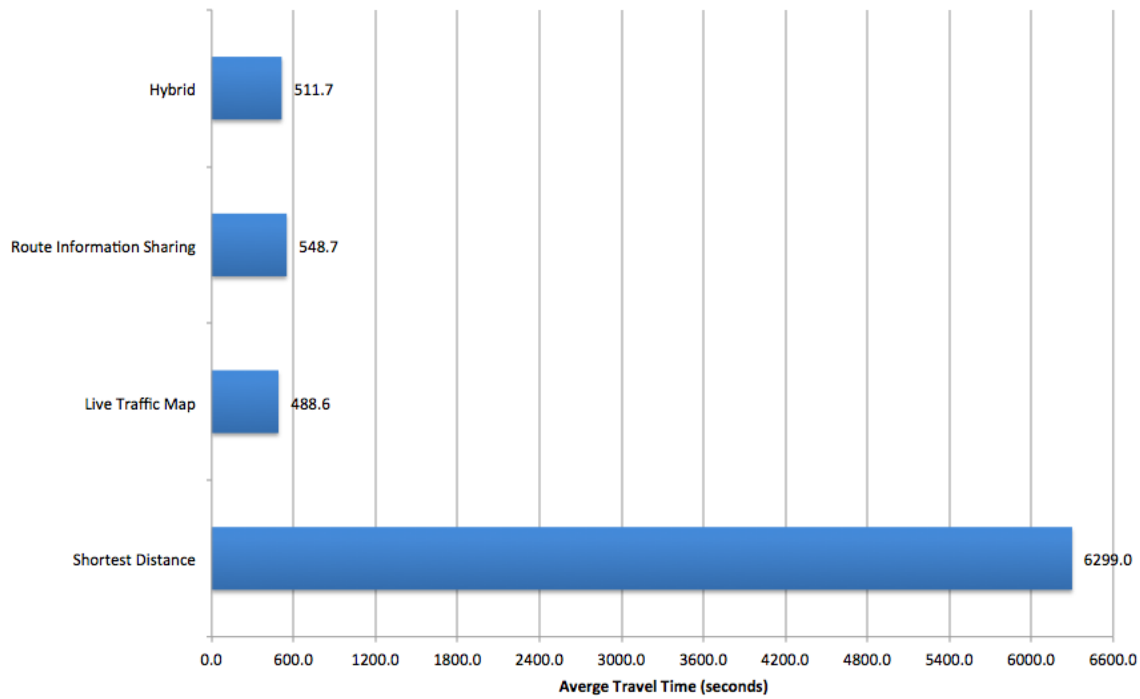

Figure 6 - Overall average travel times in Accident scenario

The poorest performer is clearly the shortest distance method, which has no knowledge of the accident and therefore does not reroute any vehicles around it. The best performer for overall average travel times was the live traffic map method. It may seem surprising that the live traffic map outperformed the route information sharing method overall, but this is due to the steady-state situation that exists for most of the scenario. Each method reroutes due to the accident differently, and therefore a closer examination of the vehicles during the simulation is necessary. Figure 7 shows the average speeds of all the vehicles at each second using the different routing methods.

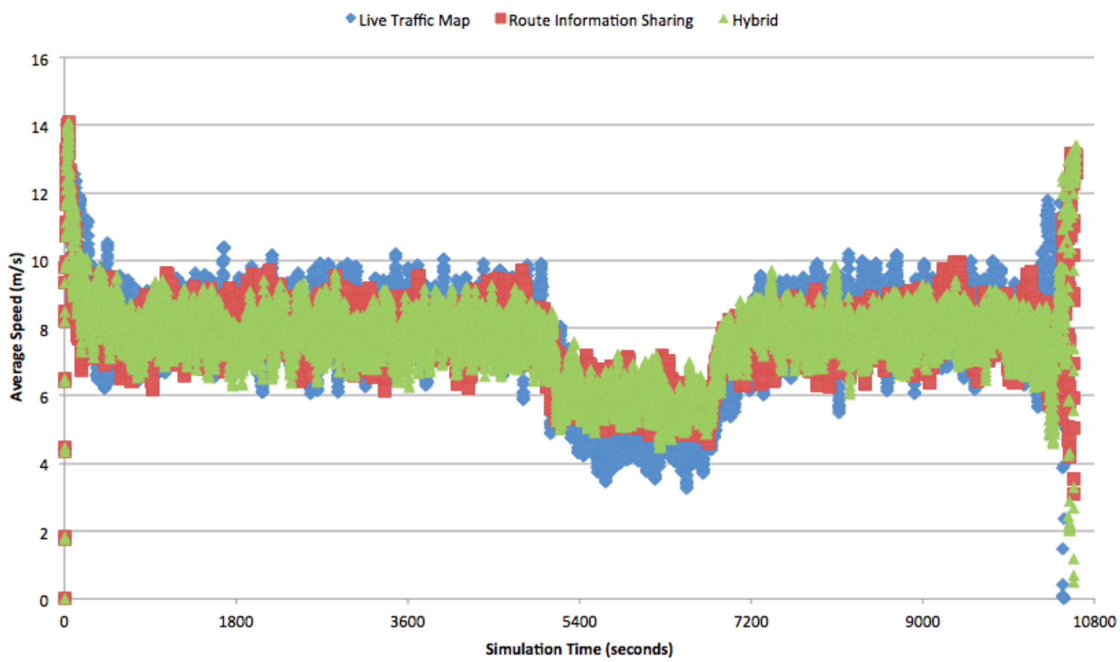

Figure 7 - Overall average speeds in Accident scenario

After the vehicles have filled the roadways, the routing algorithms effectively distributed the traffic and reach a steady state from about 1,000 seconds until the accident. During the accident, note that most of the one-second averages for the live traffic map fall below the averages for hybrid and route information sharing methods. After the accident, the original steady state is reached until the last of the vehicles are created and traffic thins out.

I would like to focus on three areas of this scenario. First, the transition from a steady state before the accident to a new steady state during the accident. Second, the time during the accident. And third, the transition from the accident steady state returning to the free-flow steady state. Figure 8 shows the average vehicle speeds around the beginning of the accident.

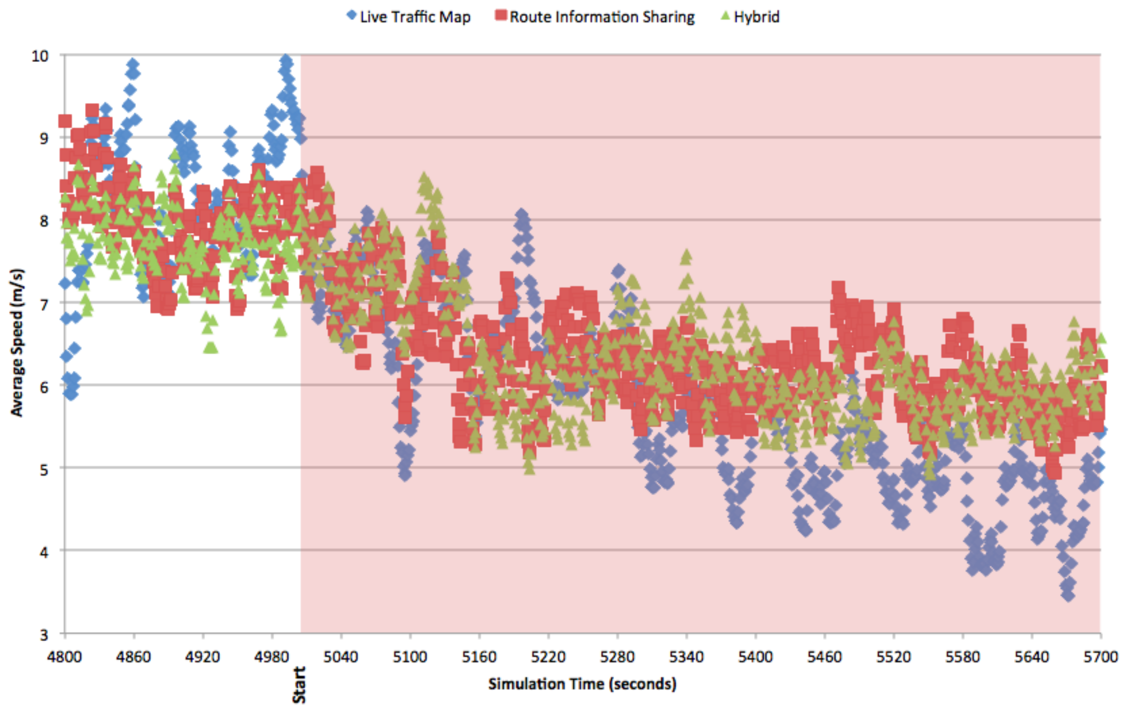

Figure 8 - Average speeds around beginning of accident

Notice that the average speeds in all of the methods begin to drop at the accident, but the route information sharing methods reach their lowest value before the live traffic map and the average speeds for the live traffic map are the lowest. The live traffic map method is capable of rerouting traffic, but it does this through many more iterations of trial and error than the route information sharing methods. The average travel time for vehicles created during this period can be seen in figure 9.

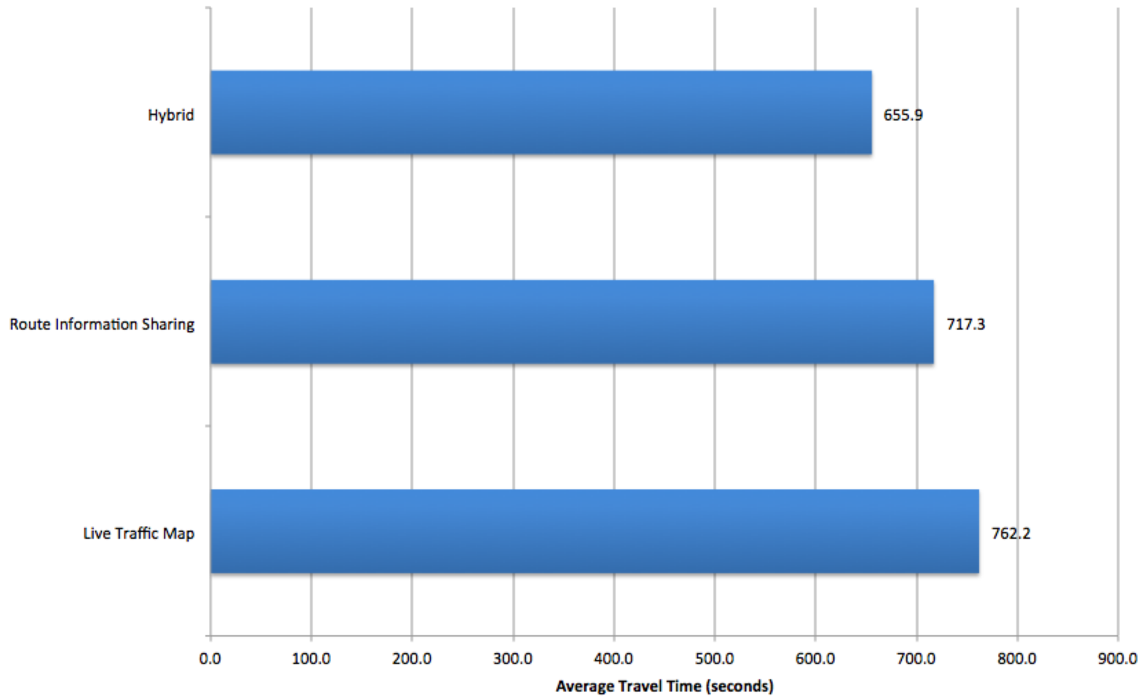

Figure 9 - Average travel times around beginning of accident

This highlights the improvement of route information sharing over the live traffic map method. The route information sharing method travel times are about 6% faster than the live traffic map method, and this cuts off almost a minute of travel time. The hybrid method may take slightly longer than route information sharing to adjust, but traffic far from the accident using the hybrid method does not switch to route information sharing, and keeps with the live traffic map method that handles its constant flow better. It improves on the traditional route information sharing method by about 8%, which is over one minute faster. The ability for cars to operate in the best modes for them allows them to maintain the best average speeds during the accident, as seen in figure 10.

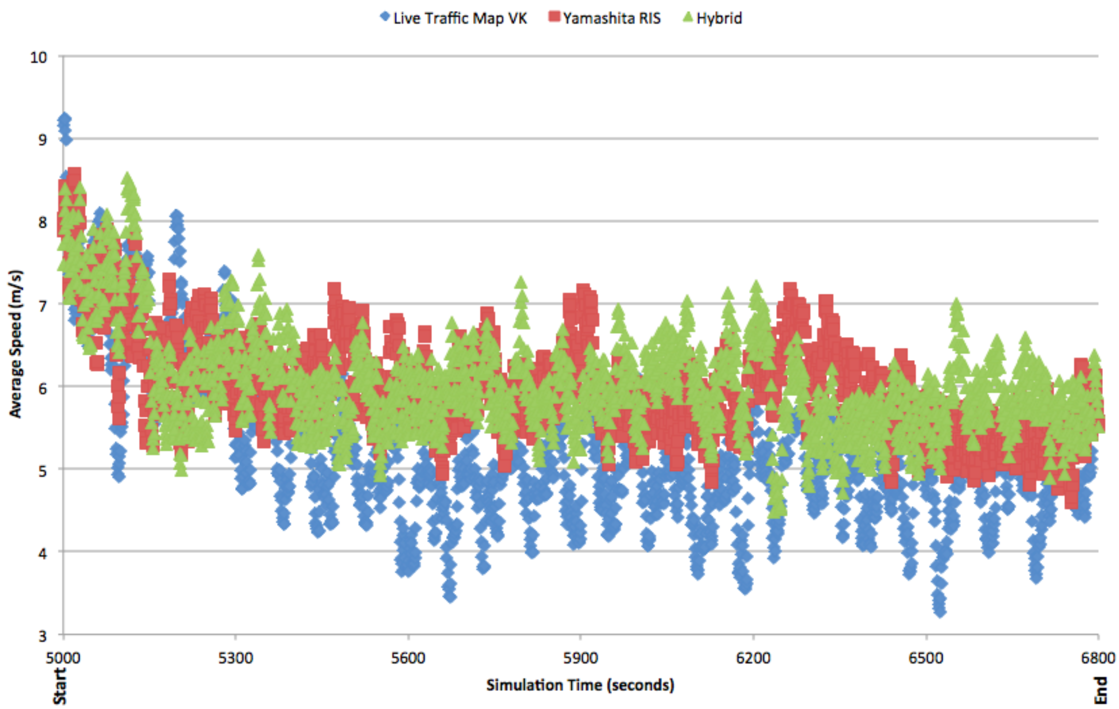

Figure 10 - Average speeds during accident

During the accident, there is a large amount of traffic taking longer paths, specifically for the vehicles whose normal path would be straight through the affected intersection. The average speeds of both route information sharing methods can be seen to handle this better than the live traffic map method. However, the scenario is working toward a new steady state, and the live traffic map method begins to catch up. The average travel times for vehicles created from the time the accident begins until it is relieved can be seen in figure 11.

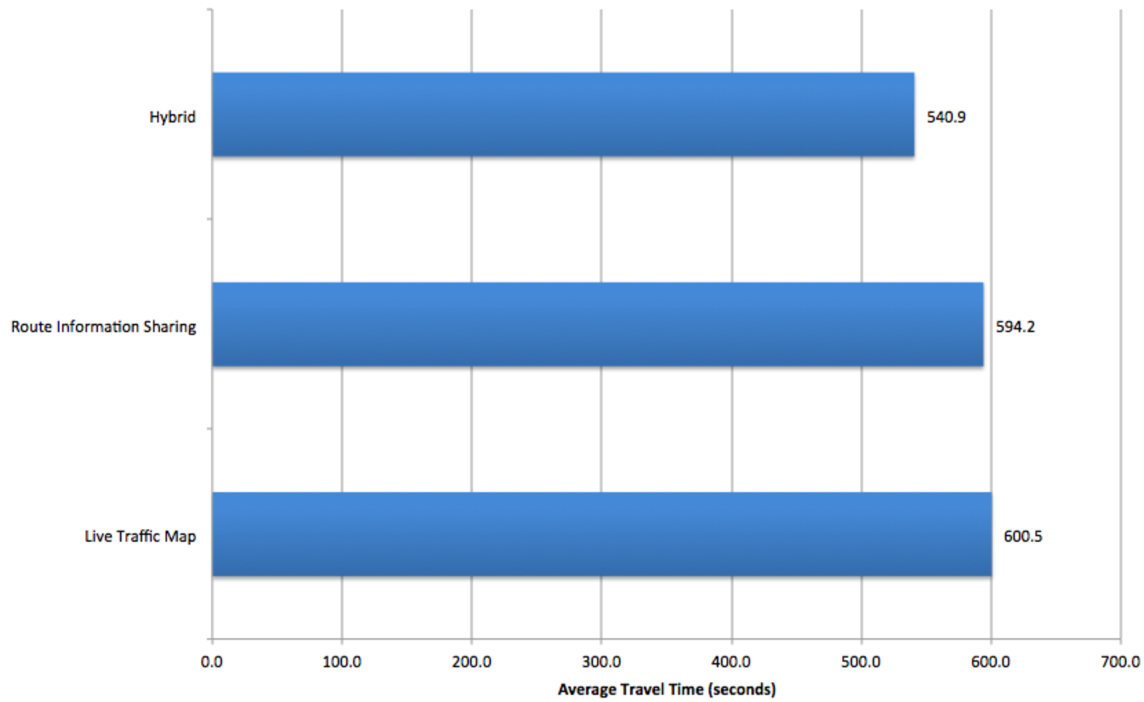

Figure 11 - Average travel times during accident

Notice that all of the travel times are lower because the traffic is reaching a new steady state. The gap has closed between the live traffic map and traditional route information sharing methods down to a one percent difference. However, the average travel times for the hybrid method are almost one minute better. At this point, a lot of the change has died down, and the few vehicles that are still facing new congestion are using the route information method to adjust while more and more are back on the live traffic map method.

The speeds as the traffic returns to normal are shown in figure 12. The reaction is similar to the one at the beginning of the accident.

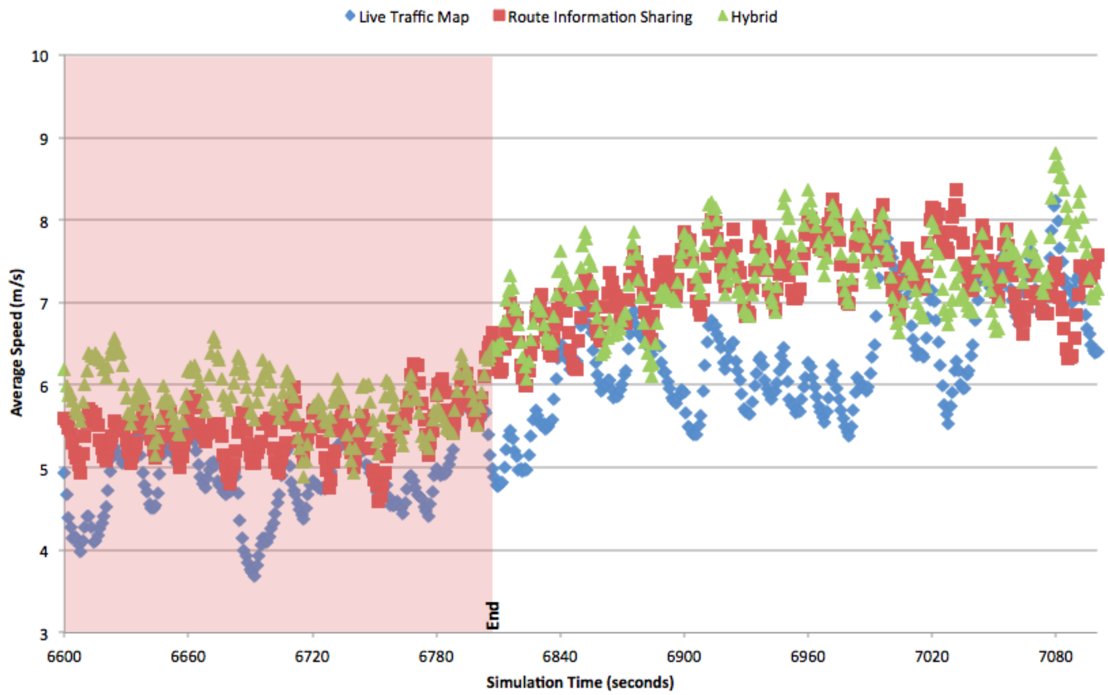

Figure 12 - Average speeds around end of accident

The route information sharing and hybrid methods are the quickest to reach the new steady state. The live traffic map method lags, but eventually reaches the highest steady state averages (as seen previously in the graph of overall average speeds for the entire scenario).

This scenario illustrates the benefits to each of the three routing methods: live traffic map excels during periods where traffic is in a steady state, route information sharing quickly distributes traffic in a scenario where traffic is changing, and the hybrid method gets a mix of benefits from both.

6.1.2 Random Scenario

The results in [14] were seen on networks using vehicles with random origin-destination pairs. Therefore, the Random scenario is worth looking at for comparison between their macroscopic simulation and my microscopic simulation. The average travel times of vehicles using each routing method are displayed in figure 13.

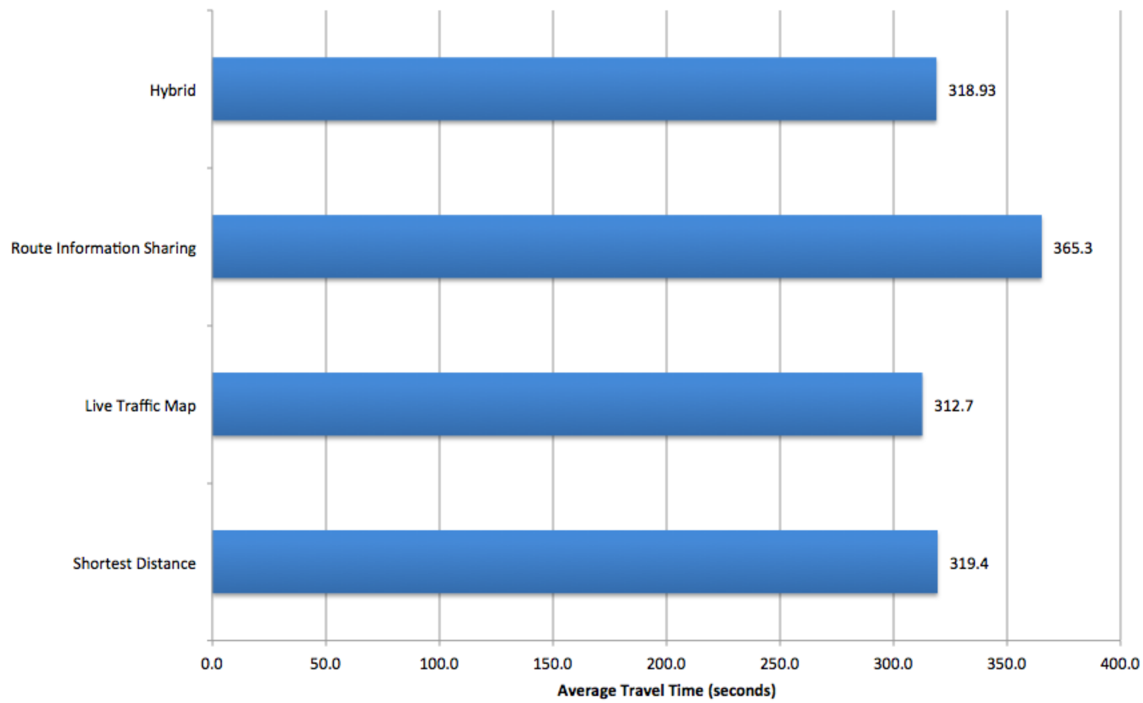

Figure 13 - Average travel times in Random scenario

The average travel times of the different routing methods are all within one minute of each other. The poorest performer was actually the route information sharing method. The hybrid method only barely beat out the shortest distance method. The route information sharing method goes out of its way to avoid predicted congestion, but in a scenario where there is random routing, it seems that a driver would be better off taking the shortest possible path, because they have a low probability of bumping into someone. The congestion levels on this random network may be low, however increasing the number of vehicles on the network would lead to an evenly distributed, steady-state congestion.

The results seen here were unexpected after seeing the results in [14], but it is worth repeating here that random origins and destinations have “nothing…to do with reality” [20]. There are many differences between the microscopic simulation in this thesis and the macroscopic simulations in [14]. The physical dimensions of the road network and vehicles in previous research were not specified; therefore it is difficult to derive an equal comparison.

6.1.3 Hurricane

A random scenario may seem like a good way to illustrate the benefits of the route information sharing method in situations of constantly changing traffic patterns, but it does not create sudden congestion. Random traffic is evenly distributed. Changing traffic patterns are not uncommon, but they usually involve a large number of cars (such as the onslaught of cars headed to the suburbs at the beginning of rush hour, away from business during lunch, or the exodus of cars at the end of a sporting event). The Hurricane scenario creates this kind of constantly changing traffic pattern. The average travel times for different routing methods can be seen in figure 14.

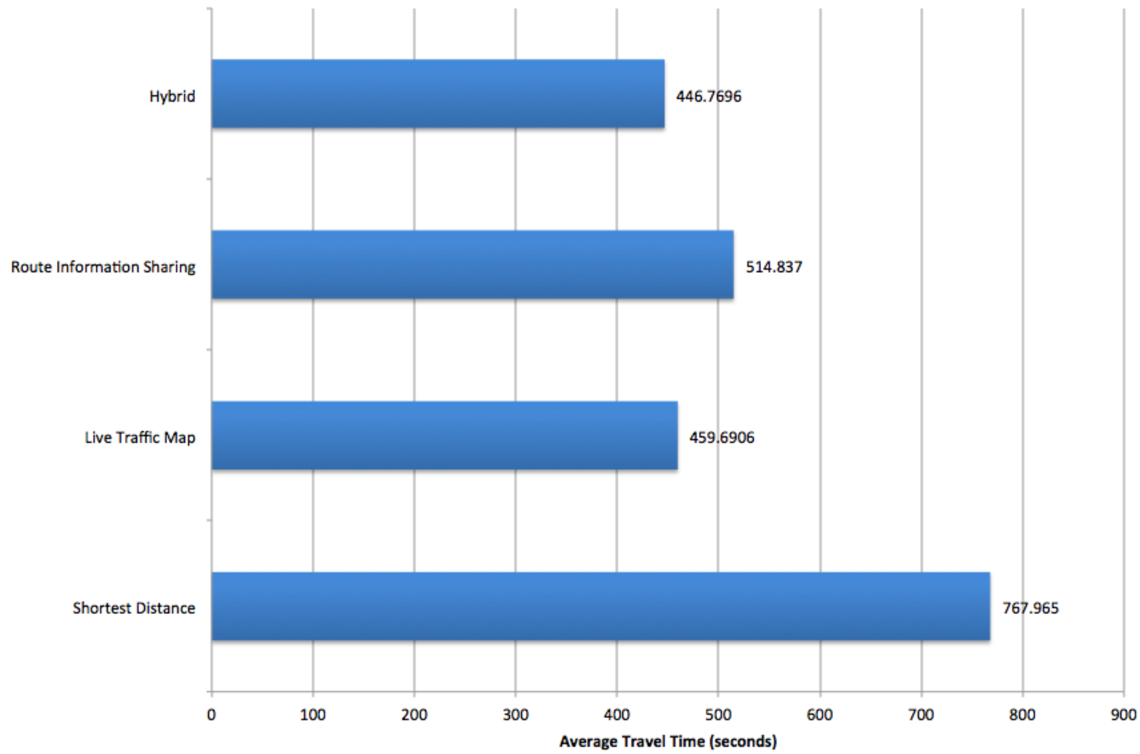

Figure 14 - Average travel times in Hurricane scenario

In this scenario, the best overall performer is the hybrid method. It does not outperform the live traffic method by much (only 2%), but it performs better than the original route information method by an average of over a minute (a 13% decrease). This simulation was repeated with different random starting and end points for the methods with communications capabilities. After fifteen simulations (five per routing method), the averages were recorded as 460, 513, and 448 seconds for live traffic map, route information sharing and hybrid, respectively. The standard deviations in these values were 14.7, 6.29, and 9.41 seconds, respectively. These results support that the hybrid and live traffic map methods perform similarly, with both out performing route information sharing.

6.2 Waiting Times

Stopping a car reduces gas mileage [1], not to mention adds wear to the break pads. In most of today's vehicles, fuel is wasted while sitting at a traffic light. New hybrid electric vehicles can turn off during these periods of stoppage. A quick examination of the waiting time characteristics of each method is also useful toward the explanation of how each method saves or adds time to a driver's commute. Waiting time is counted by SUMO whenever a vehicle's speed drops below 0.1 m/s.

The average travel times and wait times from the Accident and Hurricane scenarios are shown in figure 15 and figure 16, respectively.

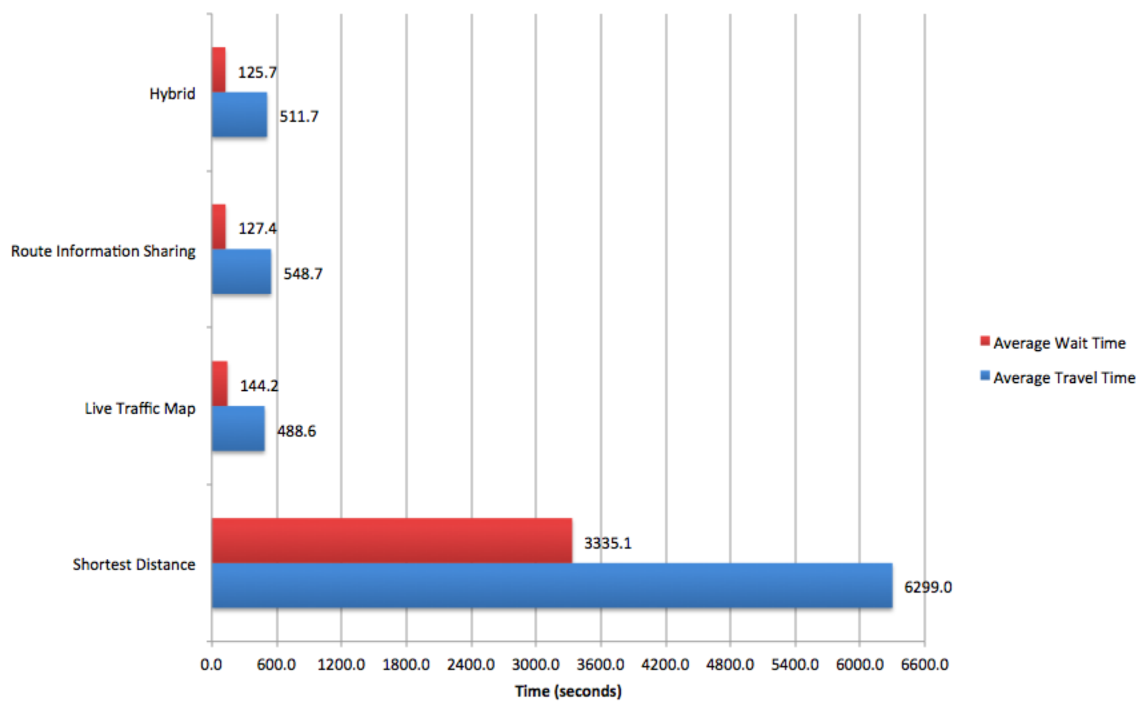

Figure 15 - Average wait and travel times for Accident scenario

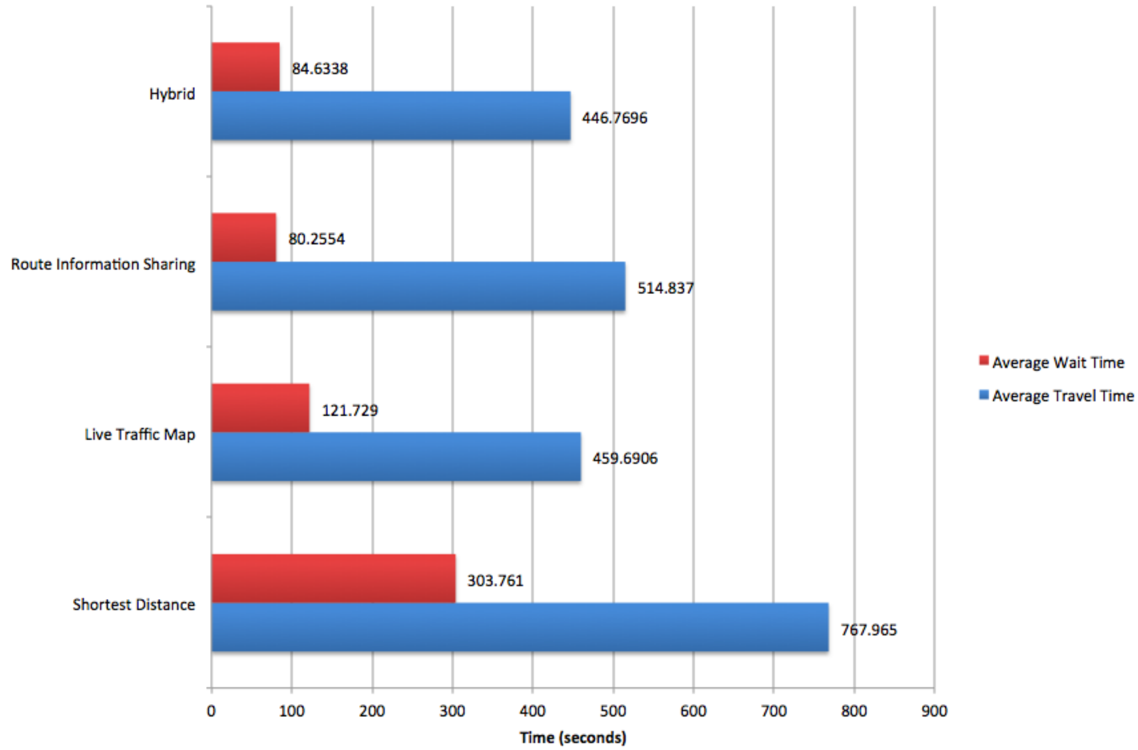

Figure 16 - Average wait and travel times for Hurricane scenario

A large amount of waiting time is experienced by vehicles routing using the shortest distance method. In the Hurricane scenario, shortest distance vehicles are stopped about 40% of their entire drive time. The waiting time in the traffic accident scenario is larger, but this average is driven up by the complete stoppage of the traffic accident. Drivers that route using the live traffic map are stopped for an average of 26% and 29% in the two scenarios. A highly touted benefit of route information sharing is avoiding congestion, and it lives up to this by having less average wait time in both scenario and a smaller percentage (only waiting for 14% and 23%) of their commutes. Staying away from congestion seems to mean more driving for the route information sharing method. The wait time for the hybrid method results in low wait times like the original route information sharing method, but still manages to have a shorter commute than route information sharing.

6.3 Hybrid Threshold Value

The hybrid routing method is meant to only use route information sharing when there is a large change in a vehicle's route, but what is a "large" change? As explained earlier, the hybrid vehicle initially determines its route using the live traffic map method and compares the number of route segments in common with the previous route. If the percentage of common route segments is below the threshold value, the route is recalculated using the route information sharing method. Therefore, the value of the threshold has an impact on the performance of this model. If the threshold is set to 0%, then the vehicle only uses the live traffic map method. If it is set to 100%, then the route information sharing method is used anytime the vehicle's path changes. Figure 17 shows data taken during a simulation of the hybrid method using a 70% threshold value in the Accident scenario. Each time the route was updated, a common segment percentage was calculated between the previous route and the live traffic map route. Any of the values in the categories below 70% caused the threshold to be broken and a route information sharing method to be calculated.

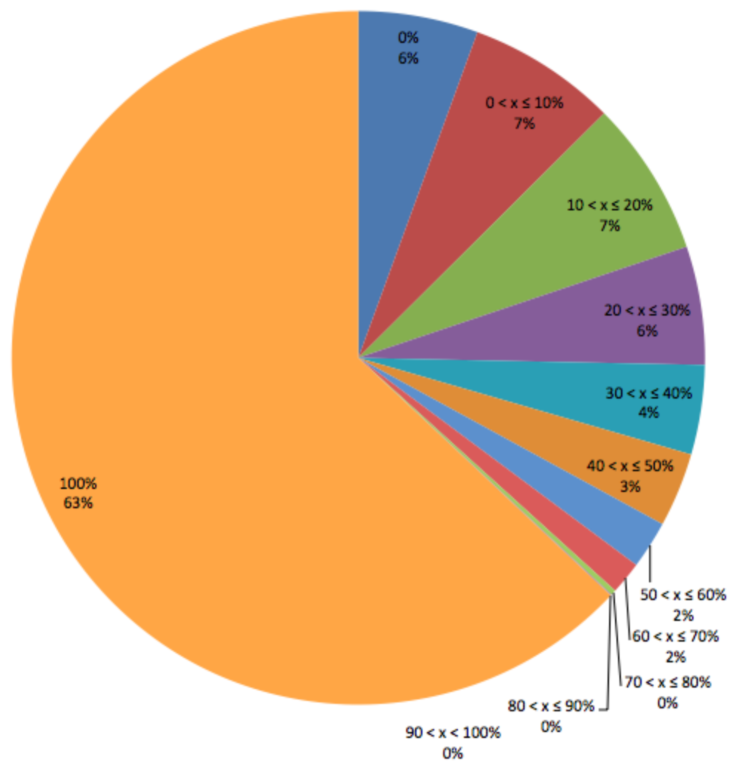

Figure 17 - Levels of common route segment percentages

The majority (63%) of the new routes were the exact same as the previous routes. There are a small percentage of new routes that had between 70% and 100% of the segments in common. This is not because this simulation had a threshold setting of 70%, rather it is due to the nature that a change in a route tends to be a major change, not a minor one. There are routes that had between 70% and 90% of their segments in common, but no comparisons had between 90% and 100% of their segments in common. This is simply mathematical. Because of the size of the road network used (21 streets wide by 21 streets long), the longest routes vehicles had were 27 route segments. In order for a vehicle to have 90% of its route segments in common, it would have had to have 25 of its 27 route segments in common. It is possible for a routing method to avoid two segments and then return to the rest of the original route, but a diversion in a route usually means a more drastic change to the route. In larger road networks, where vehicles have more route segments, a change in a vehicle's route may result in a change that retains between 90% and 100% of the route segments.

I looked across a variety of threshold values to get some insight on whether or not a particular threshold value produces better results. Figure 18 shows the overall average travel times of vehicles in the Accident scenario when using different threshold values. Note that the 20% threshold was not calculated because of a bug in SUMO.

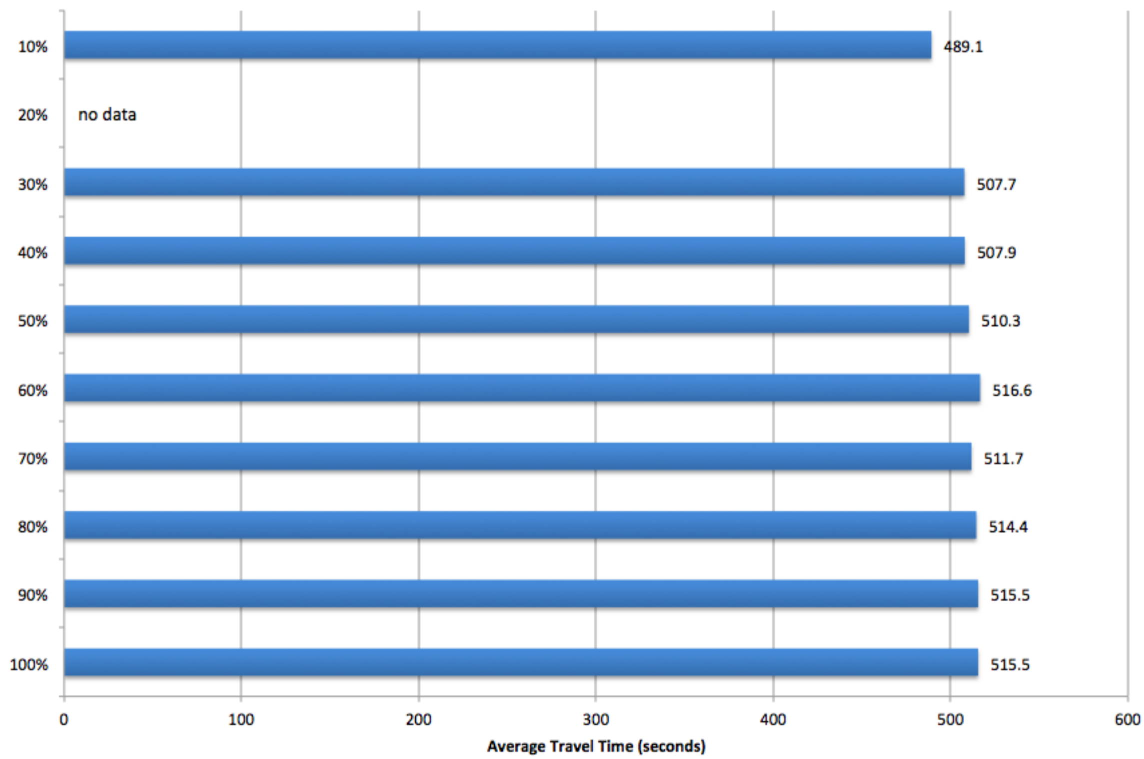

Figure 18 – Travel times across hybrid threshold values

The overall average travel times show that there is a slight trend toward better travel times as the threshold value decreases. This is consistent with our findings that the overall average travel times in this scenario are better for the live traffic map method than the route information sharing method. In fact, a threshold value of 0% would result in the sole use of the live traffic map method. However the 100% threshold does not mean that it always uses route information sharing, but anytime the route changed it would. Although there is a trend, the values themselves are all within 20 seconds of each other. This is less than 5% of their total journey. All of these thresholds outperformed using the route information sharing method alone by at least 6%. Also worth noting is a verification of the earlier accusation that a change causing between 90% and 100% of the route segments to remain in common is rarer. In fact, changing the threshold from 100% to 90% had no effect on the travel times.

The route information sharing method becomes useful when traffic patterns are changing; therefore another time worth looking at is the average travel times of vehicles entering the network around the beginning of the traffic accident. The average travel times of these vehicles can be seen in figure 19.

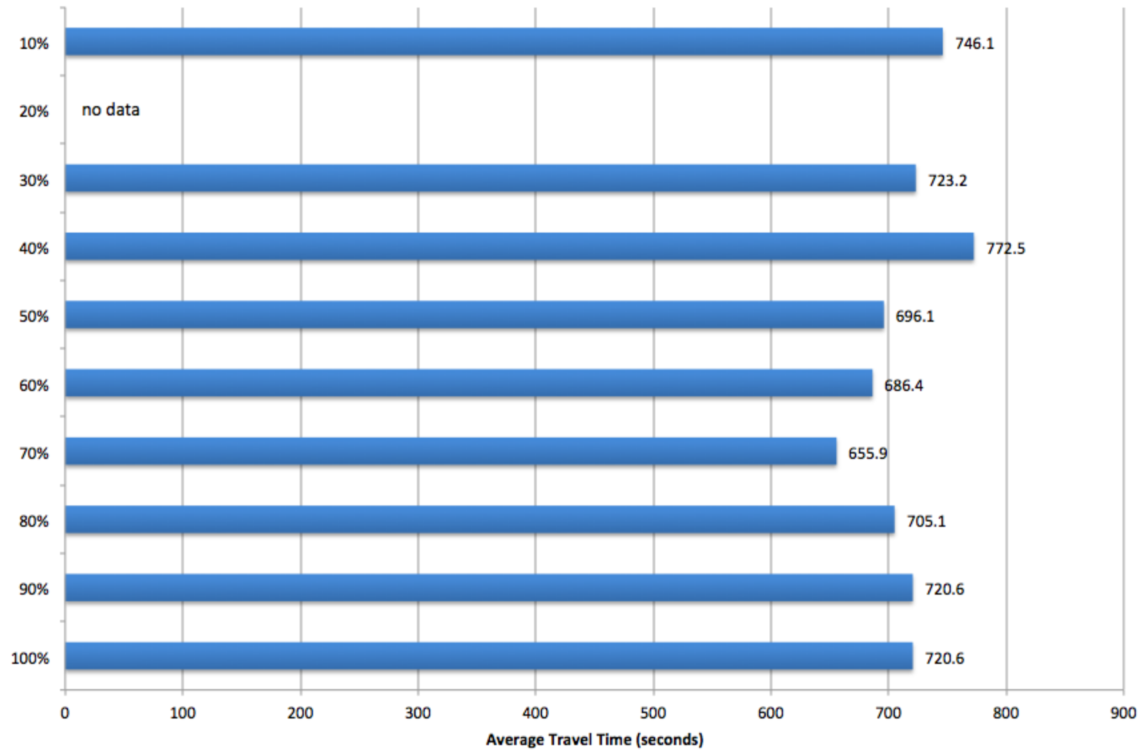

Figure 19 - Travel times across hybrid threshold values around beginning

A larger disparity can be seen between the different threshold values. The best performer was the 70% threshold value (perhaps relating to the distribution of common route segment percentages discussed earlier). This was a decrease of 15% from the poorest performing threshold value and a decrease of over 8% from using the route information method sharing alone. Again, there was no effect in changing the threshold from 100% to 90%.

A final comparison worth looking at is the effect during the end of the traffic accident. Figure 20 shows the average travel times of vehicles that entered the road network in the time span from the moment the accident occurred until it was relieved.

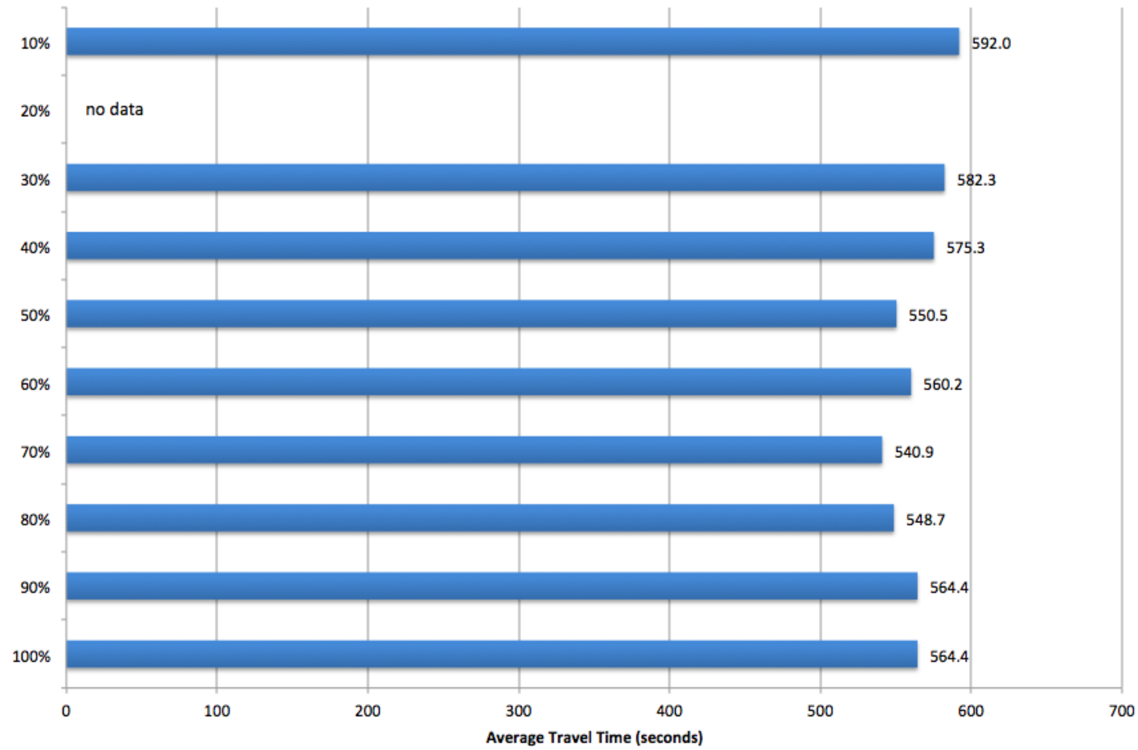

Figure 20 – Travel times across hybrid threshold values during the accident

This shows a similar trend to the travel times at the beginning of the accident. It is notable that the hybrid outperformed both the live traffic method and the route information sharing method at all of the examined threshold values.

The threshold value of 70% seems to be well suited for this scenario. This threshold value is used in the previous discussion of Hurricane and Accident scenarios average travel times. Although a threshold value of 70% may not perform well in all road networks (this may be an area for future research), it outperformed a threshold value of 100% in all of the scenarios presented in this thesis.

6.4 Information Exchange

There are plans for VANETs to use bandwidth for services in addition to traffic routing [10] [12] [13]. This may result in a limited window of time that can be used for routing. If this is the case, a broadcast rate for the routing method must be selected that serves as a compromise between bandwidth conservation and travel time performance. The following analysis is out of the assumptions that: vehicles regularly broadcast their speed and position (these simulations use once every second, though any time they enter a new road segment would be equivalent for the routing methods examined here) and route information sharing vehicles only broadcast their route when it changes. The second assumption is a simple method of bandwidth conservation. If it was not the case, then this analysis becomes much easier: expect one route broadcast per car per broadcast period. The first assumption ensures that vehicles using route information sharing can still be accounted for as the travel on their journey, without needing to completely rebroadcast their routes for passage weight recalculation when they reach a new road segment.

6.4.1 Route Change Comparison

The amount of bandwidth required for vehicle broadcasts is related to the number of route changes that need to be made. For route information sharing vehicles, there will be an initial map broadcast at the beginning of the period and there may be follow-up broadcasts of updated maps as vehicles adjust their routes and converge toward a "best solution." Figure 21 shows this for different broadcast periods and numbers of follow-up broadcasts. This graph shows the average number of route changes over the entire simulation for the different broadcast rates.

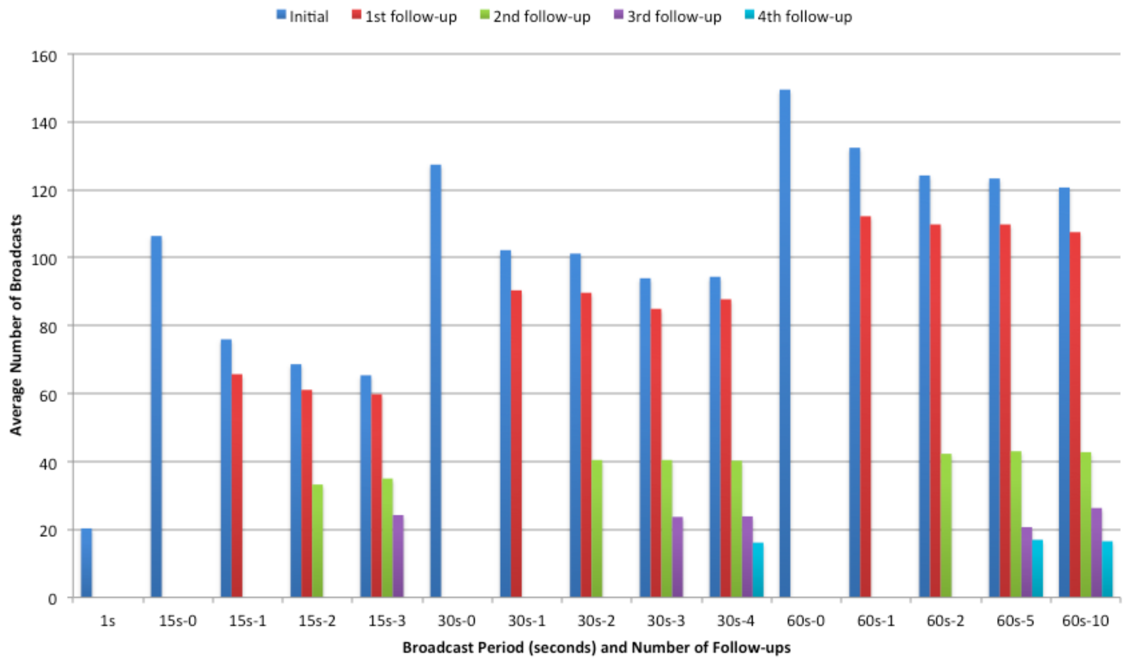

Figure 21 - Route changes based on broadcast rate

The average number of broadcasts decreases, as the broadcast period is more frequent. This makes sense because a more frequently updated map results in route updates coming as needed, rather than in large bunches.

Following the first broadcast period, the largest number of route changes is seen. Successive follow-up broadcasts result in fewer route changes for every scenario. When vehicles receive their first updated map in a while (at the beginning of the period), there are many route changes to be made. These route changes will affect other vehicles (who also choose the same path), so some of the vehicles will want to try a new route when the newly updated follow-up map is sent. As follow-ups are sent, the routes are better distributed across the road network, and fewer vehicles need to make changes. Any network of significant size will constantly have vehicles entering and leaving the roadways and vehicles reaching new road segments. Therefore, the number of route changes will never permanently reach zero. This can be seen in the figure 22, which extends out the 60 second broadcast period using 5 and 10 follow-ups.

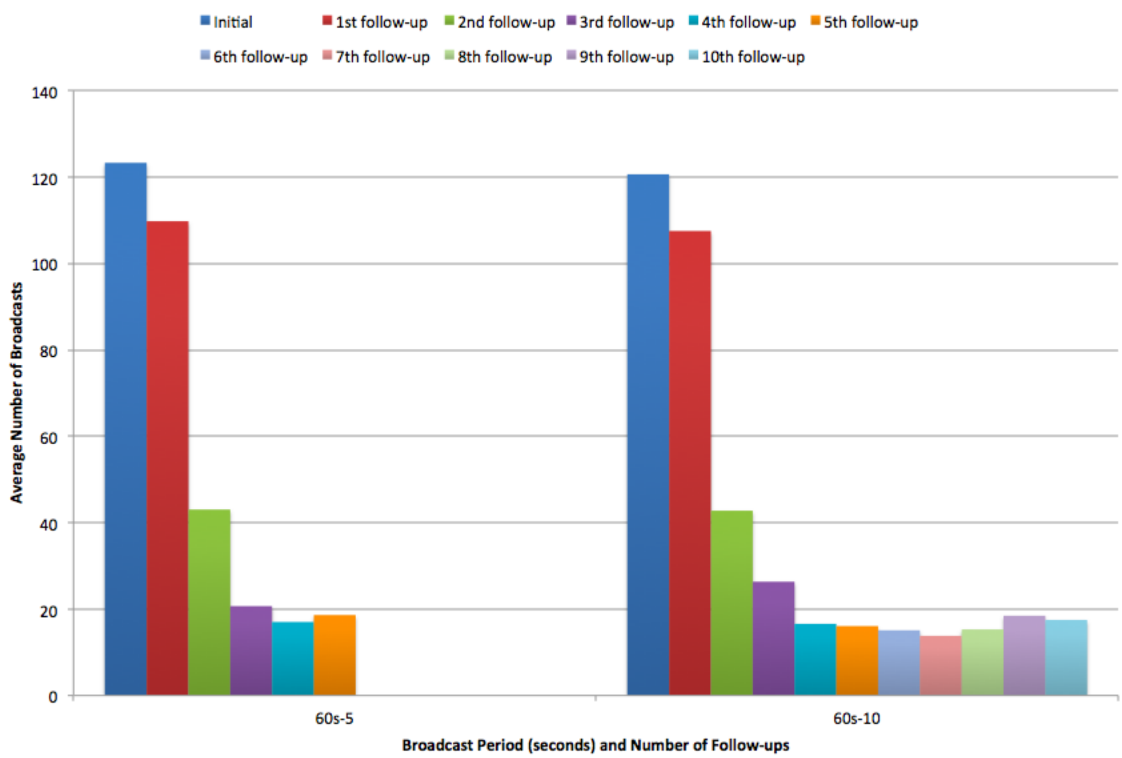

Figure 22 - Route changes for extended follow-ups

The average number of route changes settles down after about the fourth follow-up. After that, the average number of route changes does not see a significant decrease, and seems to average below 20 broadcasts per follow-up. A broadcast period of 1 second can be thought of as the case where follow-up broadcasts are allowed for the entire rest of the period.

The number of route changes throughout the simulation is shown in figure 23 for the broadcast rate of 30 seconds with four follow-ups.

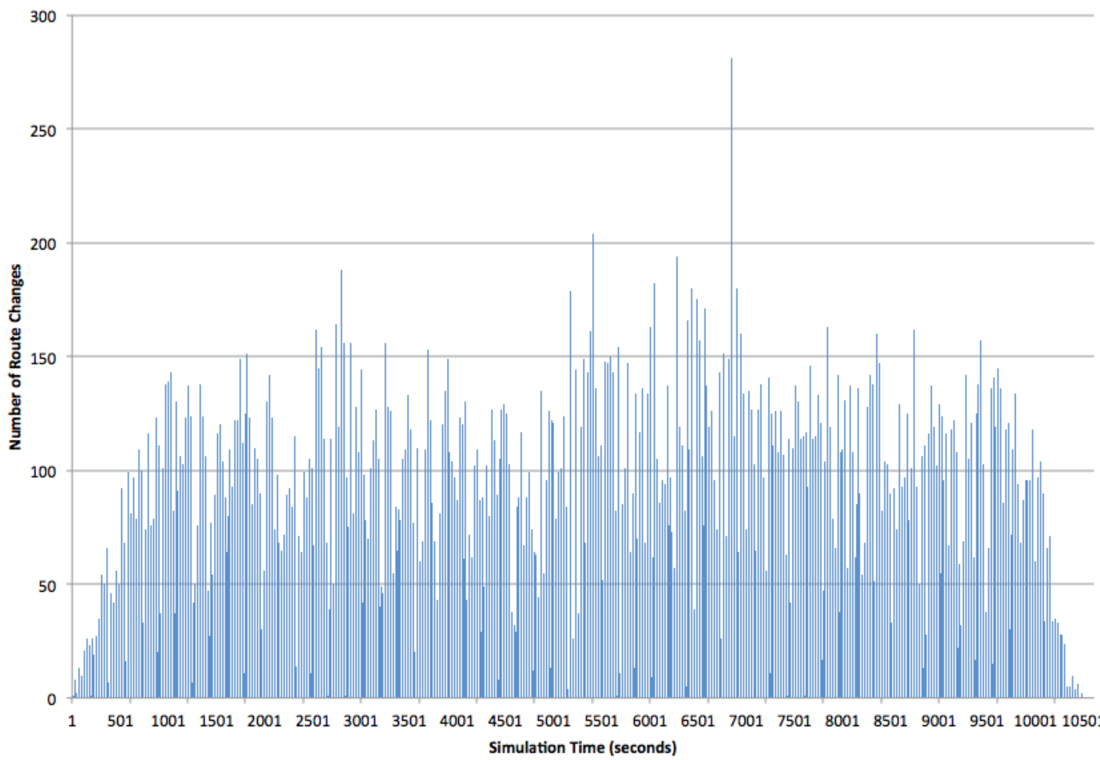

Figure 23 - Route changes at each second of the simulation

The largest number of route changes during the simulation is seen at time 6841. This peak comes forty seconds after traffic is allowed to flow again through the affected intersection. The first broadcasts after the accident is relieved would have been at time 6811 through 6815. However, it might have take a little bit of time for vehicles to begin accelerating and the traffic light go through some of its rotation before vehicles were allowed to flow again. The large peak in route changes around 6841 represents the desire of a large number of cars to take the reopened road segments (which are in the ideal route for most vehicles). The route changes at the beginning of the accident are large, but come gradually over a period of time as the roads around the accident slowly become more and more crowded.

The number of route changes is relevant to the route information sharing method, as well as the hybrid method. The hybrid method also has a broadcast period with a number of follow-ups, and does not require any more information than the route information sharing method.

6.4.2 Data Transfer

One of the goals of this research is to give estimates on the data rates that will be required of a VANET using these routing methods, so that appropriate hardware can be selected. A sample set of vehicle routes was taken from an Accident scenario simulation will be used for this comparison. The sample was taken at 7,880 seconds into the scenario and shows that there were 237 vehicles on the road network, with an average route length of 15.8 road segments. No compression is used in the calculations presented in this thesis, instead this is meant to provide a starting point for the discussion on the bandwidth costs of these methods.

6.4.3 Live Traffic Map

The live traffic map must gather data about the cars currently on the road network and broadcast the map of the overall traffic. A live traffic map using Greenshield's method only uses the number of vehicles on each road segment to predict the speed on that segment. Therefore, all that is necessary for this method is that the location of the vehicle is known so that it can be placed on a road segment.

A single GPS coordinate was shown in [23] to be 29 bits long. For latitude and longitude together, 58 bits would be needed.

The amount of data that has to be transferred is a minimum of 58 bits per vehicle. There would likely be header information, but the size of this will be dependent on the protocols of the VANET.

In order to handle the previously mentioned scenario, the vehicle network would need to be able to send 13746 bits or 1,719 bytes of data within each data gathering period.

The traffic map broadcasted in this method would be a listing of the feasible travel times for each road segment. This number should fit within a standard IEEE floating point number, which has a length of 32 bits. The protocol for doing this might simply be the list of floating point numbers, though it could be easily compressed by adding an identifier in front of each road segment and only broadcasting the road segments that vary by more than a specific amount of their free flow values. Broadcasting the entire traffic map for the accident scenario would require a minimum of 26,880 bits or 3,360 bytes. Receiving the position data and broadcasting the traffic map at this moment in the Accident scenario would have required a minimum transfer rate of about 5kBps.

6.4.3 Route Information Sharing

The data collection and map broadcast in the live traffic map method also occurs in the route information sharing method. The route information sharing map broadcast is based on a different calculation, but would still contain a number related to the estimate travel time on each road segment. Therefore, the additional data required for the route information sharing method comes from the sharing of the routes.

A route can be broadcast by sending the identifiers for the individual road segments in the route. Assuming that a road network has fewer than 4,294,967,297 road segments, the road segment identifier can fit into a single long integer (32 bits long).

Broadcasting all 3,726 road segments that were part of the routes in the Accident scenario instance would require 119232 bits or 14904 bytes. Therefore, receiving the vehicle location data, route data, and broadcasting the map would require a minimum transfer rate of approximately 20kBps. As before, this does not include the header information from the broadcast protocol, computation time, or back-off time between vehicle broadcasts.

6.5 Computation Time

The time to calculate new routes will also need to be factored into the required broadcast transfer rates. Routing methods that require more computation time will either require less frequent broadcast periods (so there is time to calculate new routes), or faster processors.

In order to provide some basis for comparison, each of the three smart routing methods was run on a modified version of the Accident scenario. Only 1000 vehicles were generated and the traffic accident was disabled. The table below shows the time it took the simulator to run each method. These times include all of the simulation time required, therefore they do not isolate the routing algorithms themselves.

Method	Total Time	Total Steps	Time per Step
Live Traffic Map	0:19:08	2507	0.457s
Route Information Sharing	0:21:06	2622	0.482s
Hybrid	0:32:38	2547	0.768s

Table 4 - Computation time

Because each simulation finishes at a different number of steps, it is important to look at the average amount of time required to go through a single step (simulated second). The live traffic map is expected to be the simplest, and it indeed requires the least amount of time. The route information sharing method requires 5% longer to run. The hybrid model runs the live traffic map method for every vehicle, but has to additionally reroute using route information sharing every time the percentage of common segments breaks the threshold. This causes it to take 60% longer than the route information sharing method.

6.6 Environmental Factors

There seems to be grant money and interest in research on ways to be more environmentally friendly and cut down on fuel consumption. These can also serve as another way of evaluating the routing methods in this thesis. The emission and fuel consumption data is produced by SUMO using the standards set in [22].

6.6.1 Vehicle Emissions

An environmentally friendly routing method would produce the smallest amounts of pollutants. The table below shows the amount of carbon dioxide, carbon monoxide, hydrocarbons, nitrogen oxides (NO and NO2), and particulate matter emitted (in kilograms) by vehicles during the entire simulation of the Accident scenario. Selected broadcast rates are also included for comparison of the effect of shortening the broadcast window.

Method	Broadcast Rate	CO2 (kg)	CO (kg)	HC (kg)	PMx (kg)	NOx (kg)
SD	N/A	22129	243.19	11.67	1.133	33.84
LTM	1s	7321	100.45	2.41	0.846	15.32
RIS	1s	8192	110.13	2.65	0.957	17.15
Hybrid	1s	7424	97.15	2.38	0.849	15.42
LTM	30s	7454	103.17	2.47	0.866	15.62
RIS	30s-4	8562	118.12	2.81	1.019	18.02
Hybrid	30s-4	7983	108.82	2.62	0.939	16.73

Table 5 - Emissions comparison

This data shows that the choice of routing method affects the amount of emissions a vehicle produces. The shortest distance method is the overall loser when it comes to emissions. Although the vehicles are taking the shortest route, their engines are running for a long time. The title of "greenest" routing method would have to be shared by the live traffic map and hybrid methods. The different wait time characteristics discussed earlier lead to different emission profiles. The live traffic map method had the lowest carbon dioxide and particulate emissions. The hybrid method had the lowest carbon monoxide, hydrocarbon, and nitrogen oxide emissions.

6.6.2 Vehicle Fuel Consumption

Everybody seems to be trying to save money at the pump. The ability for a driver to select a route based on fuel savings may not exist today, but would be worth investigating. The table below shows the fuel consumed in the Accident scenario for different routing methods and selected broadcast periods.

Method	Broadcast Rate	Fuel (gallons)
Shortest Distance	N/A	2330.75
Live Traffic Map	1s	771.08
Route Information Sharing	1s	862.80
Hybrid	1s	781.92
Live Traffic Map	30s	785.09
Route Information Sharing	30s-4 follow-ups	901.81
Hybrid	30s-4 follow-ups	840.82

Table 6 - Fuel consumption comparison

Again, the shortest distance routing method is the biggest loser. Meanwhile, the live traffic map method results in the lowest fuel consumption. The current price of gas at my local QuikTrip is $3.35. That means researchers verifying this scenario in reality would save over $5,000 by running the live traffic map method as opposed to the shortest distance method.

CHAPTER 7

CONCLUSIONS

This research is the first to use a microscopic simulation of the vehicle traffic for evaluation of a route sharing method. These simulations are based on a series of programs created around the SUMO simulator. Simulations were performed using the shortest distance, live traffic map, and route information sharing methods. Additionally, a new routing method, the hybrid method, was introduced and examined against the other routing methods. These methods were compared using the criteria of travel times, wait times, computation times, pollutants, fuel consumption, and bandwidth. This research shows that the use of live traffic information can greatly improve the efficiency of urban traffic.

7.1 Selecting a Routing Method

A goal of this research is to provide information to VANET researchers to aid them in selecting routing methods. Commute time is probably the most direct factor that a driver will notice between these methods. However, a car buyer will pay for the device that connects them to the VANET. The price of this device will be affected by the components used in it and any data cost associated with its transmission. The results in this paper indicate that the route information sharing method only outperforms the live traffic map method in situations where there are changing traffic patterns, but it carries a much higher technical cost. Based on the results in this thesis, I do not suggest the implementation of route information sharing at this time in consumer devices. Drivers using it will need faster processors, use more bandwidth, see worse fuel economy, and only rarely see the benefits. Additionally, environmentally responsible individuals would frown on its emissions when compared to other routing methods.

Route information sharing is outperformed by the live traffic map method because the live traffic map method more accurately predicts the traffic situation the car will encounter. When there is steady state traffic, the number of cars on a road segment will be the same now as the number of vehicles on the road segment when the driver reaches it. Therefore, the current situation is a very accurate predictor of the future for steady state situations. Route information sharing uses the Passage Weight to guess how much traffic will be affected by congestion. Therefore, the predictions made by the route information sharing method in steady state scenarios are not as accurate as using the current traffic situation. When the situation changes, and the amount of traffic on roadways is very different now from what it will be like when the vehicle arrives there, the live traffic map method is basing its estimated travel times only on the current information. Although the route information sharing method continues to have inaccuracies, it produces estimations closer to what will actually happen.

Before this thesis, the route information sharing method was expected to be used exclusively. However, this thesis introduced the idea of a hybrid method, where route information sharing is only used when the benefits of it would be seen. The hybrid model shares the same technical hurdles as the route information sharing method, but shows better benefits. I would suggest that this type of method be used whenever looking at implementing a route information sharing method. Based on the research in this paper, I suggest using a hybrid threshold of 70%. If other services will share the bandwidth of the routing broadcasts, then I suggest a 30 or 60 second broadcast period with four follow-up broadcasts.

I do not see conclusive enough evidence to close the door completely on pure route information sharing. In fact, I would encourage VANET researchers with equipment capable of supporting it to include route information sharing in their tests. When the technical constraints of route information sharing are not an issue, it may be a better option than live traffic map for certain scenarios.

7.2 Comparison with Results from Previous Research

I previously stated that this thesis is the first known examination of the route information sharing method at a microscopic model. The effects of traffic control, multiple lanes, different speed limits, and realistic traffic patterns seem to cause different results from those seen in [14]. This thesis should be considered as a next step from the previous research. One message that I would like to reiterate to future traffic researchers is that random traffic is unrealistic [20]. The simulations in this thesis do not perfectly model real life (no simulation can), but hopefully this thesis provides guidance for VANET researchers while future simulations are worked on.

7.3 Future Research

Several issues in this thesis were not investigated, and are suggested for future areas of research. A study on the effects of different road network shapes, specifically road networks from mapping databases, on the performance of each routing algorithm would be useful to verify these results for more realistic scenarios. The investigation into bandwidth used by each method was based on simple calculations of data from a single sample, so a more in-depth investigation into the bandwidth required for each of the routing methods using various transmission protocols would provide a more accurate picture of the required bandwidth. The methods used for calculating estimating road speeds in [14] may not be the best, and a look at other methods for estimating travel time based on the knowledge of vehicle locations and routes may outperform these. A simulation tying together SUMO and NS-2 would show effects of ad hoc message propagation problems on the routing methods. Longer simulations that include the shifting traffic patterns seen over the length of a day could give a clearer estimate of the usefulness of these routing methods.

REFERENCES

[1] Ray Magiozzi and Tom Magliozzi. Car Talk. [Online]. http://www.cartalk.com/content/features/fueleconomy/

[2] Jonathan Watts. (2009, April) guardian.co.uk. [Online]. http://www.guardian.co.uk/environment/2009/apr/06/beijing-pollution-carbon-cars

[3] James Woodcock and Aldred Rachel, "Cars, corporations, and commodities: Consequences for the social determinants of health.," *Emerging Themes in Epidemiology*, no. 17427622, pp. 1-11, Jan. 2008.

[4] Hannes Hartenstein et al., "Position-Aware Ad Hoc Wireless Networks for Inter-Vehicle Communications: the Fleetnet Project," *Proceeding MobiHoc '01 Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing* , pp. 259-262, 2001.

[5] KOTV. (2011, February) NewsOn6.com. [Online]. http://www.newson6.com/story/13962302/spring-river-rescue-in-ottawa-county?redirected=true

[6] Tamer Nadeema, Sasan Dashtinezhada, Chunyuan Liaoa, and Liviu Iftode, "TrafficView: Traffic Data Dissemination using Car-to-Car Communication," *ACM Mobile Computing and Communications Review (MC2R)*, vol. 8, no. 3, pp. 6-19, July 2004.

[8] Klemen Kozmus Trajkovski, Oskar Sterle, and Bojan Stopar, "Sturdy Positioning with High Sensitivity GPS Sensors Under Adverse Conditions," *Sensors*, vol. 11, no. 9, p. 16p.

[7] Luciano Bononi et al., "A Differentiated Distributed Coordination Function MAC Protocol for Cluster-based Wireless Ad Hoc Networks," *PE-WASUN'04*, pp. 77-86, October 2004.

[9] Lars Wischhof, André Ebner, Hermann Rohling, Matthias Lott, and Rüdiger Halfm, "Adaptive Broadcast for Travel and Traffic Information Distribution Based on Inter-Vehicle Communication," *Intelligent Vehicles Symposium*, pp. 6-11, June 2003.

[10] Tony K. Mak, Kenneth P. Laberteaux, and Raja Sengupta, "A Multi-Channel VANET Providing Concurrent Safety and Commercial Services," *VANET'05*, pp. 1-9, September 2005.

[11] Marios D. Dikaiakos, Saif Iqbal, Tamer Nadeem, and Liviu Iftode, "VITP: An Information Transfer Protocol for Vehicular Computing," *VANET'05*, pp. 30-39, September 2005.

[12] Gökhan Korkmaz, Eylem Ekici, and Füsun Özgüner, "A New High Throughput Internet Access Protocol for Vehicular Networks," *VANET'05*, pp. 89-90, Septemeber 2005.

[13] Tong Wu, *Performance Analysis And Modeling of IEEE 802.11 Based Broadcast Schemes for Emergency Message Propagation in Vehiular Ad Hoc Networks*. Tulsa, OK, USA: The University of Tulsa, 2010.

[14] Tomohisa Yamashita, Kiyoshi Izumi, Koichi Kurumatani, and Hideyuki Nakshima, *Smoth Traffic Flow with a Cooperative Car Navigation System*. Utrecht, Netherlands: AAMAS, 2005.

[15] Tomohisa Yamashita, Kiyoshi Izumi, and Koichi Kurmatani, *Investigation of Reduction of Traffic Congestion with Route Information Sharing*. New York City: AAMAS, 2004.

[16] SUMO. (2011, Feb.) Project SUMO. [Online]. http://sumo.sourceforge.net/

[18] Lester Patrick. (2005, July) Policy Almanac. [Online]. http://www.policyalmanac.org/games/aStarTutorial.htm

[17] Naugtur. (2009, January) Wikipedia. [Online]. http://en.wikipedia.org/wiki/File:AstarExample.gif

[19] KV Krishna Rao and Tom V. Mathew, *Introduction to Transportation Engineering*. Bombay: NPTEL, 2006.

[20] SUMO. Simultion of Urban Mobility. [Online]. http://sourceforge.net/apps/mediawiki/sumo/index.php?title=IntroductionDemand

[21] SUMO. (2011, March) Project SUMO. [Online]. http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Definition_of_Vehicles,_Vehicle_Types,_and_Routes

[22] INFRAS, *The Handbook of Emission Factors for Road Transport*. Bern, Switzerland: INFRAS, 2010.

[23] Annika Hinze, and George Buchanan Ryan Lever, "Compressing GPS Data on Mobile Devices," *On the Move to Meaningful Internet Systems 2006: OTM 2006*, vol. 4278, 2006.