Summer 2021

MOVE: Mobile Observers Variants and Extensions

Ryan Florin
Old Dominion University, ryan.a.florin@gmail.com

MOVE: MOBILE OBSERVERS VARIANTS AND EXTENSIONS

by

Ryan Florin
B.S. May 2005, Old Dominion University
M.S. May 2011, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2021

Approved by:

Stephan Olariu (Director)

Michele C. Weigle (Member)

Ravi Mukkamala (Member)

Dimitrie Popescu (Member)

ABSTRACT

MOVE: MOBILE OBSERVERS VARIANTS AND EXTENSIONS

Ryan Florin
Old Dominion University, 2021
Director: Dr. Stephan Olariu

Traffic state estimation is a fundamental task of Intelligent Transportation Systems. Recent advances in sensor technology and emerging computer and vehicular communications paradigms have brought the task of estimating traffic state parameters in real-time within reach.

This has led to the main research question of this thesis: Can a vehicle accurately estimate traffic parameters using onboard resources shared through CV technology in a lightweight manner without utilizing centralized or roadside infrastructure?

In 1954 Wardrop and Charlesworth proposed the Moving Observer method to measure traffic parameters based on an observed number of vehicle passes. We start by proposing methods for detecting vehicle passes using both radar and V2X as a well as with V2X only.

Next, a modernization of the Moving Observer method, called the MO1 method, using the capabilities of modern vehicles is proposed which mitigates some of the limitations of the original method. The results show our method is able to provide estimates comparable to stationary observer methods, even in low flow scenarios.

The MO2 method also utilizes two vehicles traveling in the same direction to determine a density between the two vehicles. Again, the results show this method provides estimates comparable to stationary observer methods, even in low flow scenarios.

The MO3 method is similar to the MO2 method; however, here the two vehicles travel in oncoming traffic. In doing so, the vehicles' relative velocity is large, leading us to hypothesize that the method will work well in urban traffic. The results for the MO3 method in urban traffic did not meet our expectations, which inspired us to develop the MO3-Flow method. The MO3-Flow method aggregates the counts of multiple vehicles to determine flow.

The MO3-Flow method requires additional roadside infrastructure. To remove this need, a Virtual Road Side Unit architecture is proposed. This architecture uses vehicles on the roadway to act in place of roadside infrastructure. We show this architecture provides ample service coverage if the data image is sufficiently small.

Dedicated to my wife Lauren and my community of family, friends, and church who made the last several years joyful.

ACKNOWLEDGEMENTS

I first thank my advisor, Dr. Stephan Olariu for the countless hours of dreaming of solutions to problems, pouring through data, writing papers, re-writing papers, and doing it all over again. Paper after paper, he was the driving force that kept me going.

Next, I thank my committee members, Dr. Michele C. Weigle, Dr. Ravi Mukkamala, and Dr. Dimitrie Popescu. Their feedback to this work was instrumental to its final form.

I also thank my community of friends that have been the support I have needed over the past several years. I especially thank Kenneth Findley, Chris Gammill, and Jason Kastrounis for being there to lift me up during struggles and to celebrate every win along the way.

I thank my parents for teaching me the importance of education and taking pride in my accomplishments. Additionally, I thank my father for instilling in me a love for computers and a fascination with science.

Most important of all, I thank my wife Lauren for her love and support and for bearing with me throughout this journey. Finally I thank my most cherished blessings, my children, Leah, Jenna, Andrew, Hailey, Peter, Kate, and ... for being the reason for so much happiness and silliness.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

In 2019, in the United States there were over 250 million vehicles criss-crossing four million miles of roadways [1]. Since 40% of US roadways are congested [2], congestion is a common occurrence triggered by chance fluctuations in traffic flow. Recent statistics from the US Department of Transportation (US-DOT) and National Highway Traffic Safety Administration (NHTSA) have revealed that over half of all roadway congestion is caused by traffic-related incidents and poor traffic light scheduling, rather than by recurring rush-hour traffic in big cities [3, 4]. In fact, 33% of the total delay due to congestion is experienced in non-peak times [5].

Worse yet, according to the NHTSA, congested roadways and city streets are the leading cause of tens of thousands of traffic-related fatalities [6]. In addition to the loss of life and property caused by traffic accidents, it is shown that, in 2017, congestion cost the nation over 3.3 billion gallons of wasted fuel and 8.8 billions hours of lost productivity, with these numbers expected to grow to 3.6 billion gallons of wasted fuel and 10 billion hours of lost productivity by 2025 [5]. Additionally, in 2017 individual drivers experienced an average of 54 hours of sitting in congested traffic.

There are four basic approaches to the problem of preventing congestion or mitigating its effect [7]:

1. Remove vehicles from the roadway by promoting the use of public transportation and carpooling,

2. Add new roads and lanes to provide alternate routes, thus increasing the bandwidth of the current roadways,

3. Increase average velocity by either increasing the posted speed limit or through autonomous platooning, and

4. Decrease the travel delay on the existing roadways by routing vehicles by shortest path or retiming traffic lights to optimize flow.

The last category of approaches is the motivation of this thesis. It strives to minimize travel delay on existing roads, either by routing the vehicle to the shortest path or by retiming the traffic signals to optimize flow and, thus, decrease delay. One approach to implementing this latter strategy is to employ guidance systems that provide the driving public with information on current traffic conditions, expected travel times, delays, road construction and the like. The idea is using this information, the drivers can decide for themselves what alternative best suits their needs. Systems that display travel times on overhead highway signs have been utilized in Europe and Japan for more than three decades. Recently, the high penetration of smartphones has made it possible to provide drivers with recent traffic conditions. Google Maps, 511 Traffic [8], and WAZE [9] are examples of applications that can display recent traffic conditions on a smartphone.

However, before systems can provide shortest path or retime traffic lights effectively, the traffic parameters, flow, density, and velocity, must be known by the infrastructure.

The goal of this thesis is to estimate traffic parameters in real-time with more granularity on any road and report to the Traffic Management Center (TMC). The TMC can use the data to monitor congestion, manage the traffic in real-time, and inform drivers of conditions [10]. Additionaly, the TMC can use more granular data to retime traffic lights appropriately and to better simulate the effects of changes to the roads [11].

1.2 TECHNICAL BACKGROUND

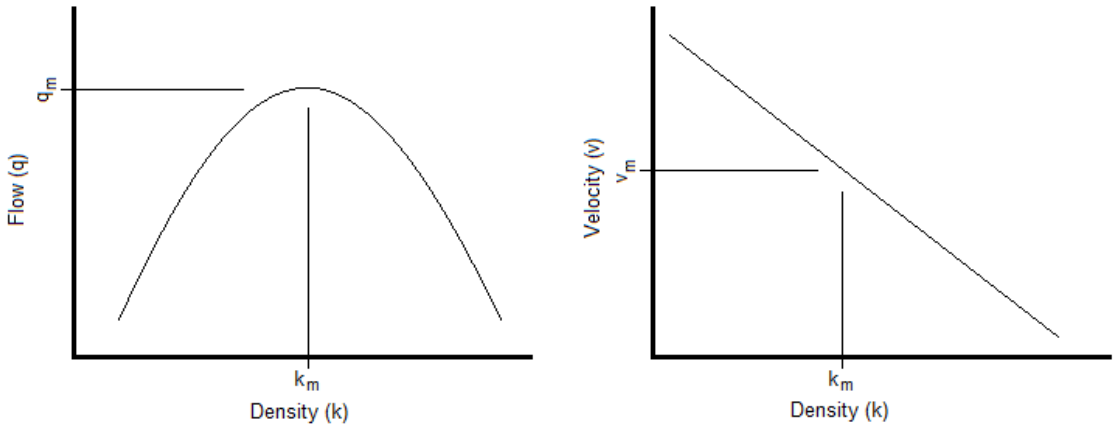
1.2.1 TRAFFIC PARAMETERS

Three fundamental parameters of traffic are flow, density, and velocity. Flow, q, is the number of vehicles that pass a certain point over a given time span. The density, κ, is the number of vehicles measured over a given distance. Finally, velocity is typically measured as an average velocity, v_{avg}, over the vehicles in a given area.

In the fixed observer method, a stationary observer placed by the roadside counts the number of passing vehicles, n, and estimates the average velocity, v_{avg}, of the vehicles passing the observer in a given time interval t. From this information, the traffic flow, q, measured in vehicles per unit time, can be evaluated as

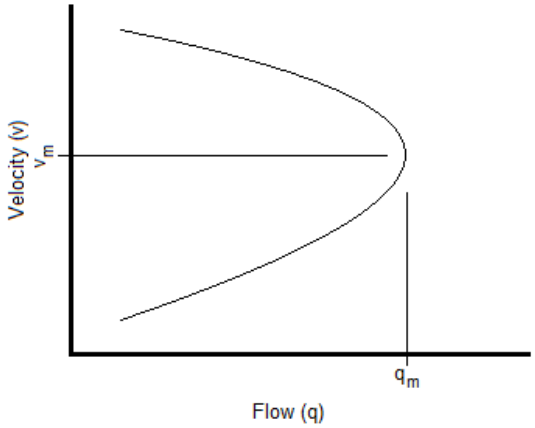
$$q = \frac{n}{t}. \tag{1}$$

In other methods, an observer is able to determine the number of vehicles, n, in a



(a) The flow - density relationship

(b) The velocity - density relationship



(c) The velocity - flow relationship

Fig. 1: The fundamental diagram of traffic flow

segment of road with length d. With n and d known, the traffic density, κ, can be measured in vehicles per unit distance using

$$\kappa = \frac{n}{d}. \tag{2}$$

Once the average velocity, v_{avg}, and either flow, q, or density, κ, is known, the other value can be estimated using the using the fundamental equation

$$q = \kappa \, v_{avg} \tag{3}$$

or equivalently,

$$\kappa = \frac{q}{v_{avg}}. \tag{4}$$

These fundamental equations are indeed so important to traffic engineers they are described in a series of diagrams known as the fundamental diagrams, see Figure 1. In figure 1(a), the relationship between flow and density is shown. As the flow increases to q_m, the max flow, the density also increases towards k_m, the max density. As the density increases past k_m, the flow will then decrease. This is because of the natural limit to the number of vehicles that can fit in the roadway. As density increases, the vehicles tend to slow down, thus leading to a lower flow. In figure 1(b), the relationship between velocity and density is shown. This is a linear relationship; as density increases, the average velocity decreases. This is because as the congestion increases, vehicles must slow down. Finally, in figure 1(c), the relationship between velocity and flow is shown. As the average velocity of vehicles increases toward v_m, the max velocity, the flow of vehicles will increase; however, if the average velocity increases past v_m, the flow of vehicles will decrease. This is because once v_m is reached, the road reaches a congested state where vehicles must slow down, resulting in lower flows.

1.2.2 TIME-SPACE DIAGRAMS

The time-space diagram is used by traffic engineers to illustrate important concepts.

In the plot, on the x-axis, the roadway will be considered one dimension and referred to as *space*. This will represent a vehicles distance from some origin point. On the y-axis, the time is plotted.

As shown in Figure 2, multiple vehicles can be plotted to illustrate key concepts.

In the figure, there are four vehicles labeled A, B, C, and D. As time increases each vehicle moves forward in this one dimensional space, or road.

Point *a.* illustrates the headway distance between two vehicles A and B. Point *b.* illustrates the difference in time that vehicle A and B pass the same point on the roadway. Where two lines cross is the position and time where the two vehicles pass on the road as illustrated by point *c.* Point *d.* illustrates the flow of the roadway at some point in the road in the interval between t_0 and t_1. The lines passing between the two points at this point represent vehicles that are included in the flow. Finally, point *e.* illustrates the density at some point on the road between x_0 and x_1. Again, the lines passing between the two points at this time represent the vehicles included in the density.

The time-space diagram is a simple way to represent the concepts, but this becomes

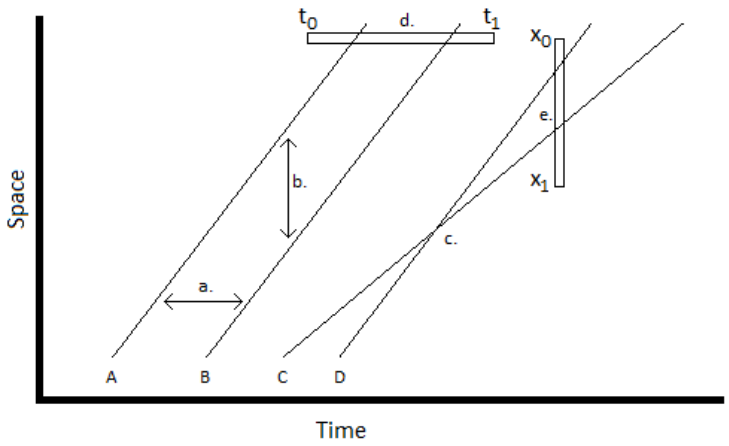


Fig. 2: Illustrating the time-space diagram.

difficult when the number of vehicles in the plot is increased, as in Figure 3. In this plot notice that the position of some vehicles increases as time increases, but also the position of some vehicles decreases as time increases. This represents vehicles traveling the road in both directions. Also, notice that the times are not all straight. This represents that the vehicles' velocities are not constant.

1.2.3 CONSERVATION OF FLOW

The conservation of flow is an important principle in traffic engineering that describes that vehicles will not be created or destroyed in unnatural ways. For example, consider a one-way stretch of road starting at point A and ending at point B, with no addition entrances or exits. Simply, the conservation of flow tells us that a vehicle can enter only at A and can exit only at B.

Additionally, consider the same one-way stretch of road and an interval I where $I = [t_1, t_2]$. If a vehicle enters the stretch of road at point A at the beginning of the interval, then there are only two options for where it can be at the end of the interval. The vehicle may be between point A and point B, or it may have already passed point B at some time prior to t_2.

1.2.4 TRAFFIC STATE ESTIMATION

Traffic state estimation is of fundamental importance to ITS since it plays a key role in

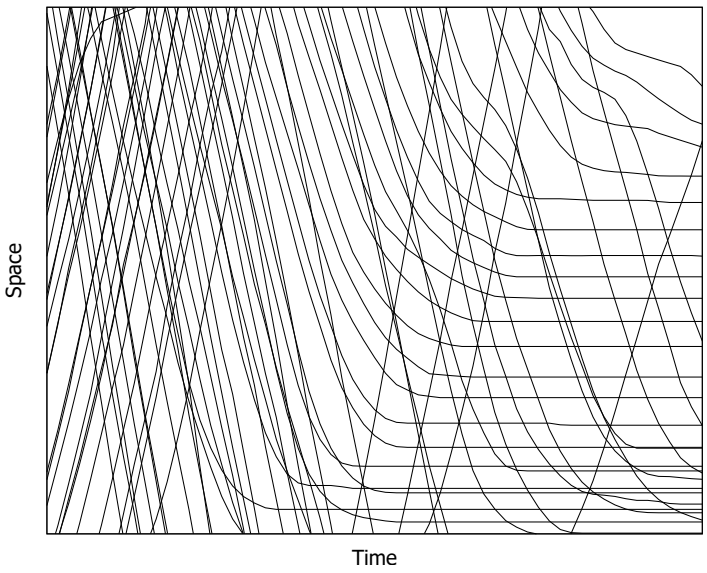


Fig. 3: Illustrating a realistic time-space diagram.

dynamic route guidance, incident detection, short-term travel time prediction, and various Measures of Effectiveness (MOE) parameters [12, 13]. Not surprisingly, the topic has received a good deal of well-deserved attention in the recent literature [14, 15]. We refer to Seo *et al.* [16] for a comprehensive survey of traffic state estimating strategies.

Over the years, two broad classes of strategies have been developed to solve this problem. In the first such strategy, known as the *stationary observer* (SO) method, stationary observers (e.g., pneumatic tubes, inductive loop detectors, cameras, microwave sensors, radars) placed by the roadside count the number of vehicles passing the observer in a given time interval and, having estimated their average velocity, evaluate traffic density [13, 17].

Several researchers have pointed out that the SO method, and its myriad variants, yield accurate traffic state estimates but are quite expensive to field and maintain. In spite of this, the estimates obtained by using the SO method are considered by many authors as benchmark values against which the accuracy of all other methods is assessed [16].

Legacy traffic monitoring and incident detection techniques, which are still in widespread use today, employ such stationary observers: inductive loop detectors (ILD), video detection systems, acoustic tracking systems and microwave radar sensors [18, 19]. By far the most prevalent are the ILDs, which are placed on the roadway every mile or so. The ILDs measure traffic flow by registering a signal each time a vehicle passes over them. Each ILD, including

hardware and controllers, costs around $8,200; in addition, adjacent ILDs are connected by optical fiber that costs $300,000 per mile [20]. It is well documented [21] that the legacy equipment installed in support of collecting traffic-related data is expensive and costly to maintain and repair. Not surprisingly, transportation departments worldwide are looking for less expensive and more reliable solutions for traffic monitoring and incident detection [22].

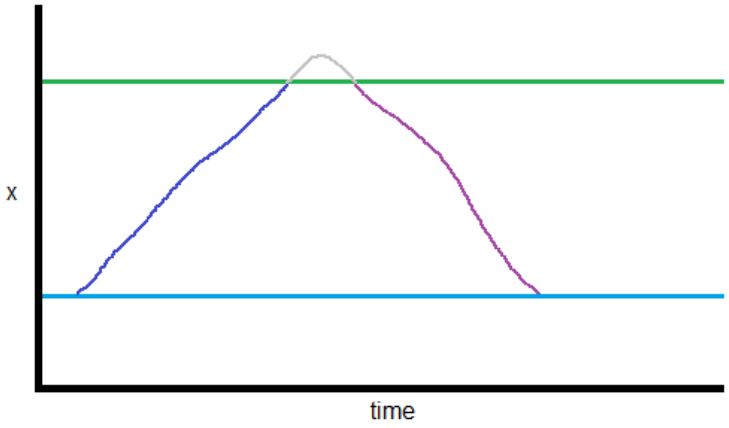
To be effective, innovative traffic-event detection systems may need to enlist the help of the most recent technological advances. For example, recent advances in sensor technology have produced cement-based piezoelectric sensors that do not corrode, cannot be damaged by thermal expansion of the road, and can be made of inexpensive materials [23]. These sensors can be embedded in the roadway and detect vehicles like ILDs. They have been the basis of the NOTICE system [24] that involves embedding intelligent sensor belts in roadways and using these belts to detect traffic-related events ranging from congestion to lane obstructions and potholes. NOTICE has a great deal in common with ILDs since both systems are intrusive and contribute to weakening the structural integrity of roadways. Extrapolating from past experience with ILDs, sensor belts embedded in the roadway are very likely to suffer from reliability problems and to contribute to the creation of potholes.

An idea that exploits the prevalence of smartphones is to supplement legacy traffic monitoring with traffic incident reports submitted by the driving public. A recent implementation of this idea has lead to 511 Traffic that offers an at-a-glance view of road conditions in a given geographic area [8]. Unfortunately, 511 Traffic is a centralized system that accumulates and aggregates traffic-related feeds at Traffic Monitoring Centers and, due to inherent delays, often displays stale traffic information [25].

1.2.5 MOVING OBSERVER METHOD

In 1954, Wardrop and Charlesworth [26] proposed a moving observer method for estimating traffic density, flow, and velocity. The method employed a *driver* and a human vehicle *counter*. Use Figure 4 as an illustration of the method. The driver will drive down the stretch of road being measured then turn around and drive back to the starting point. When driving with codirectional traffic, the counter will keep tallies for the number of times another vehicle is passed by the test vehicle, n_s, and the number of times the test vehicle passes another vehicle, n_f. Additionally the counter will run a stop watch to keep track of the time, t, it takes to travel the stretch of roadway. Finally, the counter will also make note of the distance traveled, l. When the driver turns around, the counter will count the

(a) Roadway view



(b) Time-space view

Fig. 4: Illustrating the original Wardrop and Charlesworth method.

number of vehicles passed in the oncoming traffic, n'_f, and run a stop watch keeping track of the time, t', it takes to get back to the starting point. The distance, l, is the same for both directions.

The flow, q, and density, κ, can be estimated using the following equations:

$$q = \frac{l'(n_f - n_s) - l(n'_f - n'_s)}{l't - lt'}. \tag{5}$$

and

$$\kappa = \frac{t'(n_f - n_s) - t(n'_f - n'_s)}{l't - lt'}. \tag{6}$$

Note that while traveling the route, the results may be biased by the velocity of the test vehicle. As a result, the test vehicle may implement a strategy known as the *Floating Vehicle* method [27, 28], in which the velocity of the test vehicle is calibrated in such a way that it passes and is passed by an equal number of vehicles.

As several authors have pointed out, the drawback of the moving observer method is

that in order to achieve results comparable to those obtained by the stationary observer method, the test vehicle has to perform a significant number of runs in both directions [27, 28, 29]. Unfortunately, all this takes time during which the traffic parameters may well change. Yet another shortcoming of the moving observer method is that it works only when the test vehicle traveling in the opposite direction can count all the vehicles encountered in the direction of interest. While this works well on two-lane country roads, is may not work well on multi-lane roadways where the median presents obstacles that preclude the test vehicle from obtaining an accurate head count of the vehicles in the oncoming direction [27, 29].

Finally, the moving observer method does not take into account the reality of present-day vehicles, equipped with DSRC-compliant radio transceivers [30], cameras [31, 32], and an assortment of short- and long-range on-board radar devices [33].

Florin and Olariu [34] have proposed an enhanced Moving Observer method that uses the on-board capabilities of present-day vehicles and that achieves a level of accuracy comparable to that afforded by the SO method. This work is included in Chapter 4.

Quite recently, Florin and Olariu [35] have proposed a variant of the Moving Observer Method that uses the tallies of two passing vehicles that also achieves highly accurate results. This work is included in Chapter 5.

1.2.6 MODERN VEHICLES

It has been the case for many years for vehicles to include an on-board computer and a GPS device coupled with a digital map, whereas higher-end vehicles have many more features and technology; this is no longer the case. New vehicles today are being equipped with a plentiful array of technology with the intent of keeping the driver, passenger, and other vehicles safe on the roads. Lidar is being used by experimental autonomous driving vehicles, but the cost of such systems is prohibiting its use in stock vehicles. Radar and cameras are being used as a cheaper alternative to lidar [36, 37]. Additionally, radar is being used for Adaptive Cruise Control, forward collision warnings, Blind Spot Information System (BLIS), and other incoming vehicle warnings [38].

As an example of these sensors provided by manufacturers is the Ford Co-Pilot 360 [39] that is offered in many of their models. It includes adaptive cruise control using radar and lane centering and lane exit warnings using cameras. Some additional packages also include active parking assistance, which uses ultrasonic sensors to park the vehicle on behalf of the driver.

Other types of sensors include outside air temperature sensors, rain sensors, humidity sensors, traction control sensors, tire pressure sensors, in addition to many more that handle the inner workings of the engine. Altogether, the vehicle is highly aware of how it is working to make it more efficient, but also highly aware of its environment. More and more, vehicles are also being equipped with a radio transceiver to allow for cooperation between other vehicles. The sensors of today's vehicles are being used for the driver of the vehicle in which they are installed; however, tomorrow's vehicles will share this information to allow cooperative sensing of the road.

1.2.7 CONNECTED VEHICLES

Recently, inspired by the increasing sophistication of present-day vehicles and motivated by advances in vehicular networking, the US-DOT has started promoting the Connected Vehicles (CV) initiative [40]. By using dedicated wireless connectivity between the vehicles participating in the traffic, the CV initiative aims to promote an increased awareness of real-time traffic conditions and to reduce the number and severity of crashes [41]. It was started under the name Vehicle Infrastructure Integration (VII), with the name changing to Intellidrive, and finally to Connected Vehicle. CV is a top priority of ITS with the focus on adoption of technologies and future deployment [42].

It is estimated that CV can mitigate over 80 percent of non-impaired incidents. Increased cooperation between vehicles and infrastructure allows for safety applications such as collision avoidance, violation warnings, and basic information (ie. sharp curve warnings) [43].

As it turns out, one of the benefits of the CV initiative is that it enables distributed data collection and traffic state estimation. Importantly, these operations are performed locally by the vehicles involved in the traffic flow, rather than remotely in the cloud. This saves time without sacrificing computational accuracy, a definite plus. Finally, the CV technology ensures that the dissemination of the results occurs in real-time using some flavor of Vehicle to Vehicle (V2V) and/or Vehicle to Infrastructure (V2I) communications.

1.2.8 V2X

Vehicle to Everything (V2X) is the collective term for the set of standards used within vehicular networks to share safety, informational, and entertainment based messages. Two subsets of V2X, V2V and V2I, are the two vehicular communication systems that are the

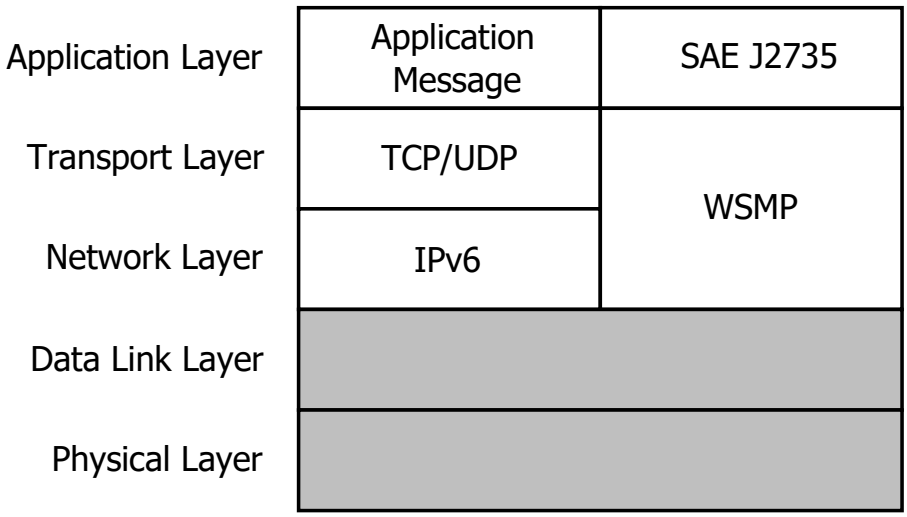


Fig. 5: Illustrating the V2X communication model.

most frequently references in literature. V2V is wireless communication between two vehicles; it primary use is for safety applications such as locating typically non-visible vehicles [44]. V2I is wireless communication between a vehicle and infrastructure. This infrastructure is installed near the roadway and its main purpose is to provide information from the TMC to the vehicles, and information about the vehicle back to the TMC.

Of main importance is safety messages that include vehicle location, velocity, acceleration, and other data several times a second. These messages are transmitted over a restricted safety channel; alternatively, non-safety based messages, ie. restaurant advertisements can also be transmitted over the network, but not on the safety channel.

There are two main technologies that are competing to provide V2X communication. They are DSRC/WAVE based on IEEE 802.11p and 3GPP C-V2X. The differences in these two technologies is at the physical and data link layers of the V2X communication model. Both use the same application, network, and transport layers [45, 46]. See Figure 5 for a representation of the V2X communication model.

Starting at the network layer, there are two types of message protocols: WAVE Short Message Protocol (WSMP) and IPv6. Those which are single hop typically use WSMP which was introduced for efficient one hop transmission. Messages sent using IPV6 typically

require the routing that IPV6 has to offer, and are typically multi-hop [47].

The single hop messages using WSMP send messages that are defined in SAE J2735 [48]. These messages, most importantly, includes the Basic Safety Message (BSM), but also includes other vehicle application messages as well.

The other types of messages sent via IPv6 utilize either TCP or UDP in the Transport Layer. These are application specific messages.

In this thesis, the differences at the Physical and MAC layers are not relevant; instead of relying of a particular technology, the more generic term *V2X* will be used.

1.2.9 DSRC/WAVE

Dedicated Short Range Communications for Wireless Access in Vehicular Environments, typically referenced as DSRC/WAVE is based on IEEE 802.11p.

DSRC enabled devices may communicate to a set of devices that all agree to exchange data together, called a Basic Service Set (BSS). There are two types, infrastructure and independent BSS. The infrastructure BSS will have an access point, typically an RSU, that is connected to a distribution system (DS). The independent BSS has no access point or distribution system, but is instead a standalone device. Prior to sending data to either type of BSS, the device must first join. The AP or independent device will send a BSS announcement, after which other devices may join.

The protocol also allows for messages to be send outside the context of a BSS (OCB). OCB messages do not require devices to join prior to sending; this is ideal in vehicle communication as this process adds overhead which may be difficult in highly mobile scenarios. Devices that are part of a BSS may not send OCB messages.

Services available are announced on the Control Channel (CCH). On the Service Channel (SCH), only WAVE Service Advertisement (WSA) messages and other management messages can be sent. The WSA messages include the service advertisement and the SCH to access the service. There are certain intervals defined where the device should listen in the SCH or the CCH.

Basic Safety Messages are sent OCB on a dedicated channel, which is "exclusively for vehicle-to-vehicle safety communications for accident avoidance and mitigation, and safety of life and property applications" [49]. Data sent in these packets include location, heading, velocity, acceleration, elevation, and other such data needed by safety applications.

1.2.10 C-V2X

Cellular-V2X (C-V2X), first defined in 3GPP Release 14 [50], is the second and more recent of the two V2X communication technologies. It was developed to make use of the existing mature cellphone technology. C-V2X allows communication to the cell phone tower through the use of the LTE interface, which offers a distinct advantage over DSRC.

In addition to the ability to communicate with a cell phone tower, it includes the ability to use a sidelink to communicate directly to another vehicle without the use of a cell phone tower. This *sidelink* is made available using the PC5 interface. Similar to DSRC, the services are announced on a Physical sidelink control channel (PSCCH), and the services are accessed on one of the Physical sidelink shared channel (PSSCH).

Recently, in Novemeber 2020, the use of the 5.9GHz spectrum has been allocated to C-V2X [51]; however, the C-V2X specification are still being finalized. In this thesis, some DSRC standards are assumed, given that DSRC is the more mature; however, it is expected that similar standards will be set for C-V2X in the future.

1.2.11 V2X MESSAGE PROTOCOLS

A set of messages designed for V2X is described in SAE J2735 [48]. The messages in this standard are intended for use by DSRC; however, have also been adopted by C-V2X [45].

Basic Safety Messages

The Basic Safety Message (BSM) is the single most important message for vehicular communications. It is a message that is sent by each vehicle ten times a second that includes information including position, velocity, and heading. Its intent is to be used to track locations of vehicles to increase the safety of roadways.

The BSM has the following format that is also shown in Table 1:

Header - The header will identify the type of message. The format is described in more detail in the Header Format.

Part I - Part I includes required fields including position, velocity, and heading. This information is sent with each BSM message. The format is described in more details in the Part I Format.

Part II - Part II of the message are optional fields and are sent less frequently. This part includes events, vehicle status, and local content. Part II is not used in this thesis and is not described in detail.

This completes the BSM message format.

Next is the BSM Header which is also described in Table 1(b):

TABLE 1: BSM format and subfields

(a) The BSM Format

Field	required	size
Header	required	1 byte
Part I	required	37 bytes
Part II	optional	variable bytes

(b) The Header Format

Field	required	size
Message ID	required	1 byte

(c) Part I Format

Field	required	size
Message Count	required	1 byte
Temporary ID	required	4 bytes
Second Mark	required	2 bytes
Latitude	required	4 bytes
Longitude	required	4 bytes
Elevation	required	2 bytes
Positional Accuracy	required	4 bytes
Velocity	required	2 bytes
Heading	required	2 bytes
Acceleration Set	required	7 bytes
Brake Status	required	2 bytes
Vehicle Size	required	3 bytes

Message ID - The Message ID identifies the type of message that is being sent. In this case, the value will be 1 to signify the message as a Basic Safety Message.

This completes the Header format.

Next is the Part I format, which is 37 bytes and is also described in Table 1(c):

Message Count - The Message Count field is a single byte integer from 0 to 127. It is incremented for each message of the same type. Its purpose is to determine if a message was missed. For example, if a vehicle Y retrieves a BSM message from vehicle X with a Message Count of 1, then another with a Message Count of 3, then it knows the message with Message Count 2 was missed.

Temporary ID - The Temporary ID field is a 4 byte identifier that resembles a MAC/IP

address. It may be updated regularly according to a timer, a particular event, or perhaps when the the vehicle is started. Its intention is to protect the privacy of the vehicle. Its main purpose is to act as a unique identifier that can be used to track the location of a vehicle.

Second Mark - The Second Mark field represents the millisecond within the second. It can be a value from 0 to 61,000.

Latitude - The Latitude field, which represents the actual latitude of the vehicle, is the first of the four position fields. It is 4 bytes and has a precision of 1/8th mirco degree.

Longitude - The Longitude field, which represents the actual longitude of the vehicle, is the second of the four position fields. It is also 4 bytes and has a precision of 1/8th mirco degree.

Elevation - The Elevation is the third of the four position fields. The field represents the elevation of the vehicle. It is 2 bytes and can represent -409.5 meters to 6,143.9 meters.

Positional Accuracy - The Positional Accuracy field is the final position field. It is 4 bytes and is used to determine the accuracy of the position by axis.

Velocity - The Velocity field, which represents the current velocity of the vehicle, is the first of the motion fields. It is 2 bytes and is in units of .01 m/s. It can represent a velocity of 0 to 327.65 m/s (731.5mph).

Heading - The Heading field, which represents the current heading of the vehicle, is the second of the motion fields. It is 2 bytes and each value represents a unit of 360/32,768 degrees.

Acceleration Set - The Acceleration Set field is broken into three subfields for a total of 7 bytes. The longitudinal acceleration and latitudinal acceleration fields are in units of 0.01 m/s^2 and can have values of -2,000 to 2,000. The Vertical acceleration field is in units of 0.08 m/s^2 and can have values of -127 to 127. Finally, the Yaw Rate field measures the amount of rotation in the directional of travel in units of .01 degrees per second.

Brake Status - The Brake Status field is 2 bytes and includes several subfields that represent information about the brakes. These fields include if the brake is applied, the state of traction control, the status of anti-lock brakes, the status of stability control, and if brake boost is applied.

Vehicle Size - The Vehicle Size field is 3 bytes long and includes the Vehicle Width and Vehicle Length fields, both of which are in units of centimeters.

This completes the Part I format.

Finally given that Part II is not utilized in this work, it is described only in briefly. Part

II, is optional and includes event flags and vehicle status. Event Flags is a bit string with a standard set of bits indicating an event. These include loss of control, hard breaking, and airbag deployment. When these flags are set, there are other fields in the vehicle status that will be included. Vehicle Status contains several optional fields which represents any possible information from vehicle sensors. This can include the status of wipers and external lights, air temperature and rain sensors, vehicle mass, vehicle bumper height, past location of the vehicle, and many more fields.

WAVE Service Advertisement

The WAVE Service Announcement is a Service Announcement defined by DSRC/WAVE. It is expected a similar message will be standardized with C-V2X. This service announcement message is set by the RSU on the control channel to provide an advertisement for a particular service.

TABLE 2: WSA Format

Field	required	size
WAVE Version / Change count	required	1 byte
Extension Fields	optional	variable bytes
Service Info	optional	variable bytes
Channel Info	optional	variable bytes
WAVE Routing Advertisement Fields	optional	variable bytes

The WSA Format includes the following fields that are also shown in Table 2:
WAVE Version - The WAVE Version is the version of the WSA as defined in the current version of the standard. This field is the first four bits of the first byte of the message.
Change count - The Change count field is the last four bits of the first byte of the message. It it incremented each time the WSA message changes for the same sender. It is used to recognize changes in the WSA message.

Extension Fields - The Extension Fields includes a number of fields to extend the message protocol. The possible extension field include the rate the WSA message is sent, the transmissions power used, the location, and string identifier. These fields are not used in this work and are not described in further detail.

Service Info - The Service Info field contains a list of up to 32 service advertisements. It includes the details of the service including the service type via PSID and, optionally, the IPv6 details of the service if the messages will be sent via IPv6. The Service Info Format is described in greater detail below.

Channel Info - The Channel Info field contains a list of channel information fields, each matching to a Service Info record. It includes details of the channel including the channel number, data rate, and transmit power. The Channel Info Format is described in greater detail below.

WAVE Routing Advertisement - The WAVE Routing Advertisement includes information for connecting to the Internet. It includes information such as the gateway address, DNS address, and IPv6 subnet prefix. These fields are not used in this work and are not described in further detail.

This completes the WAVE Service Advertisement format. The Service Info and Channel Info formats will now be briefly discussed.

The Service Info format includes a number of fields important to communication. The PSID is included to represent the type of service. Additionally, if it is important for the service host to receive messages via IPv6, the Service Info Extension fields will include the IPv6 address of the host, and the service port of the listening service process. Finally, the Service Info and Channel Info will be linked by a Channel Index field.

The Channel Info format most importantly includes the Channel Number of any of the host's WSM messages. Additional supporting fields are included but are not important for this thesis.

WAVE Short Message Protocol

In V2V communications, messages are sent vehicle to vehicle, also known as one hop, as opposed to being routed through multiple vehicles, or multiple hops. The WAVE Short Message Protocol (WSMP) is designed specifically to make efficient use of the wireless channel. As such the protocol has an overhead of 5 to 17 bytes. This can be compared to the UDP/IPv6 overhead of at least 52 bytes. The WSMP is used by both DSRC and C-V2X.

TABLE 3: WSM Format

Field	required	size
Version	required	1 byte
PSID	required	1-4 byte
Channel Number	optional	3 byte
Data Rate	optional	3 byte
Transmit Power Used	optional	3 byte
WSM Wave Element ID	required	1 byte
Length	required	2 byte

A WAVE Short Message (WSM) is the term used to describe a message sent using WSMP. The WSM message must be associated to a particular service by using a proper Provider Service Identifier (PSID) that is maintained by the IEEE 1609 working group.

The WSM has the following format and is also described in Table 3:

Version - The Version field represents the current version and its value is determined by the standard. The field is one byte.

PSID - The Provider Service IDentifier (PSID) identifies the service. The PSID field is a variable length identifier depending on the leading bits in the first byte. For example, if the leading bit is a 0, then the length of the PSID will be 1 byte. If the leading bits are 1110, then the PSID will be 4 bytes.

Channel Number - The Channel Number field is optional and is 3 bytes long. It represents the channel number used to communicate WSM messages.

Data Rate - The Data Rate field is optional and 3 bytes long. It represents the actual data rate used by the service host.

Transmit Power Used - The Transmit Power Used field is also optional and 3 bytes long. It represents the transmit power of the message as sent by the service host.

WSM Wave Element ID - The WSM Wave Element ID defines the format of the data payload. It is required and is 1 byte long. It also is used to mark the end of the optional fields.

Length - The Length field is the length, in bytes, of the data payload. The length field is

2 bytes.

This completes the WSM format.

IPV6 with TCP or UDP

In addition to communicating using WSM, vehicles may also communicate using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) over IPv6.

Internet Protocol Version 6 (IPv6), is simply the sixth version of the Internet Protocol which is the requirements for addressing and routing on the internet. The IP header includes, among other fields, the Source and Destination IP Address.

TCP is a connection-oriented service which requires a three-way handshake between two processes resulting in a connection between the sockets of the two processes. Additionally, TCP is also a reliable data transfer service, meaning all the data will be delivered in order and without error [52].

UDP is a simpler protocol that requires no connection, and therefore no handshake. UDP is also unreliable, in that there is no guarantee the message will be delivered [52].

The IP protocol can be seen as a means of sending the message to the correct computer; hence having the source and destination IP Address in the header. The TCP and UDP protocols can be seen as a means to routing the message to the correct process on the computer; in the TCP and UDP headers, the source and destination ports are included.

In V2X networks, TCP and UDP over IPv6 works in the same manner as it does with other networks.

1.2.12 VEHICULAR NETWORKS

There are two main categories of vehicular networks, static networks and mobile networks, based on the movement of the vehicles.

In static vehicular networks, vehicles are not moving, or the relative distance between vehicles is constant. These types of vehicular networks generally mimic traditional networks, despite being vehicles. For example, the static vehicular clouds in [53, 54, 55, 56, 57, 58] mimic conventional clouds in their architecture.

In mobile vehicular networks, the vehicle movement is dynamic. Note, that at certain moments the relative change in distance between vehicles on the road may be very small, leading to a situation where a mobile network can temporarily assume the properties of a static vehicle network. In these situations, vehicles may work together and share information back and forth for long periods of time. In other situations the relative change in

distance between vehicles may seem to be chaotic. Vehicles may still work together and share information back and forth, but they only do so for short periods of time. When mobility is low, there is the chance for redundancy of computing, data, and an increase of shared resources between a given set of vehicles. When mobility is high, vehicles may be treated as data mules, disseminating data throughout the network quickly, but resource sharing becomes less likely.

In a mobile vehicle networks, vehicles use a combination of V2V and V2I communications. While V2V communications are "zero-infrastructure", V2I communication requires a roadside unit (RSU) that acts as a gateway that supports the communication between vehicles and, say, the Internet. It is well known that RSUs are expensive to deploy and maintain and so, not surprisingly, very few urban communities have deployed them [19].

However, there are many possible ways of implementing the functionality of an RSU without physically deploying RSUs at great expense. For example, parked vehicles could play to role of RSUs and so are, to some extent, vehicles in the public transportation fleet, taxi cabs that criss-cross our streets, or indeed, police cruisers [59]. This concept will be further discussed in Chapter 7.

1.2.13 VIRTUALIZATION

Virtualization is the concept of creating a virtual version of some object. The concept started with mainframes by creating Virtual Machines (VM), or virtual environments for users, so it appeared to the user that they had their own dedicated machine. This concept later became a key component of cloud computing by allowing a VM to be allocated for the user [60].

The VM runs within a Virtual Machine Manager (VMM) that translates the operations of the VM to the hardware of the system. The VMM provides the VM the resources it needs to complete its tasks, including memory and CPU time. In this way, the resources used by the VM are well isolated from other Virtual Machines. Note that multiple VMs may be running within a VMM. Additionally, the VM can be paused and moved from one VMM to another running on a different computer. This process is known as a VM migration.

Virtualization through Virtual Machines as been proposed as a way to run user jobs on vehicles. One of the first papers to recognize fault tolerance and availability as being important in such vehicular networks is He *et al.* [61]. This work is continued by Ghazizadeh [62] who has suggested using vehicles to run jobs on virtual machines in vehicles in parking lots. Ghazizadeh proposed a number of job scheduling algorithms using redundancy to

ensure the user job completes reliably.

Refaat *et al.* [63] and Baron *et al.* [64] proposed virtual machine migration in vehicles to keep the VM within the coverage of a cloud area. Despite Baron *et al.* [64] showing it can be done even with the VM sizes of 200-400MB in some scenarios, it is suggested that because of these large VM sizes an alternative method must be found. VMs are a powerful tool, but they are large in size because they include the the operating system, the user application, and the application data.

Given the large size of VMs, the concept of Container Virtualization offers a lightweight solution [65]. In this virtualization architecture, a container runs on top of the operating system. Then within the container is an image that contains just the libraries and application needed by the user. To migrate, this smaller image can be moved to another computer with the ability to run the container.

The lightweight virtualization such as that provided by containers are more suitable for vehicle application that require migration [66]. Additionally, in Morabito *et al.* [67], lightweight virtualization is suggested as an enabled technology for smart vehicles by utilizing containers to for implementing vehicle applications.

Container virtualization is more attractive than virtual machines because there is less overhead in the virtual image size; however, with container virtualization, the host must have the appropriate container to run the virtual image.

1.3 CONTRIBUTIONS

One of the goals of the CV initiative is the development of simple and easy to implement strategies for real-time traffic state estimation. Aligned with this goal, the main contribution of this thesis is to propose simple and easy to implement real-time protocols for traffic parameter estimation.

As we see it, the main advantages of our traffic parameter estimation methods are their simplicity, ease of computation, the fact that they only need V2V communications to aggregate estimates, and that it is privacy-aware. Indeed, what sets our methods apart from other methods is that while the vast majority of the estimation methods known to us require aggregating probe vehicle data with data obtained from various flavors of stationary sensors, our method is self-sufficient in the sense that such aggregations are not necessary.

In contrast to other methods, e.g. Wardrop and Charlesworth [26] and van Erp *et al.* [14], both of which require constant flow and vehicles with constant velocity, our method assumes realistic traffic with non-constant velocities and non-constant density.

The basic idea of our methods is this: by using their on-board sensing capabilities, vehicles maintain a *tally* of the difference between the number of times other vehicles pass them and the number of times they pass other vehicles. Notice that since vehicles may vary their velocity as they please, they may pass and be passed by the same vehicle multiple times and, consequently, maintaining a correct tally is a non-trivial task. We show that the tallies computed by vehicles relate in an interesting way to traffic density and flow. A significant result, of an independent interest, is the proof of this theorem which does not rely on a certain order of vehicles in traffic, does not rely on constant velocity, and does not rely on a first-in-first-out assumption. One further point is of interest: tallies are collected in a *privacy-preserving* fashion. Indeed, we do not read license plates or other attributes of passing or passed vehicles that would uniquely identify the vehicle.

Once collected, tallies are aggregated by V2V communications between vehicles. Local traffic parameter estimates thus obtained are then disseminated using V2V communications, a suitably-decentralized network such as NOTICE [24], or some form of more conventional cellular or V2I communication.

It was our original intention to use time-space diagrams to develop and to prove the correctness of our method; however, simplifications to the traffic model would have been required. For a single vehicle the time-space diagram is a useful tool to visualizing the mobility characteristics of the vehicle. However, the time-space diagram can easily get overwhelming and unmanageable when the trajectories of multiple vehicles moving at non-constant velocity are plotted together. It is, no doubt, for this reason that when the trajectories of several vehicles are plotted on the same diagram, the simplifying assumption is made that the vehicles maintain constant velocity.

Others have sidestepped this assumption by replacing it with their own to give more or less the same result. Since the 1960s, traffic engineers have been using cumulative curves to analyze traffic phenomena [68]. Makigami *et al.* [69] relates these to time-space diagrams by adding a third dimension which tracks the cumulative count of vehicles passing by a fixed point over time. By plotting only the cumulative count and time on a graph, the cumulative curve ignores vehicle trajectories and instead plots the order of a vehicle based on a reference vehicle; in their own words, "two cars 'bounce off' each other, and exchange numbers". Seo and Kusakabe [70] use this simplification and add the more restrictive assumption that traffic adheres to a first-in-first out rule to simplify the proof of their method for traffic state estimate.

While these simplifying assumptions are harmless for the purpose of evaluating traffic

flow and density, they become unwieldy for the purpose of counting the number of times a given vehicle has passed, or was passed by other vehicles, in a given time interval. The difficulty is that since vehicular velocities are not constant, the same pair of vehicles may pass each other any number of times. It is for this reason that we cannot use time-space diagrams for proving our strategy and we had to devise a completely different mechanism that, we believe, may be of interest in its own right.

1.4 RESEARCH QUESTION

In this introduction, we have seen the effects of congestion and suggested that knowledge of traffic parameters, namely flow, density, and velocity, are necessary information for the TMC to offer more effective routing and re-timing of traffic signals. We have seen that the most popular method of measuring these currently is through stationary observer method, which means it must be installed and maintained for a steep cost.

Next we covered several topics, including the moving observer method, connected vehicles, modern vehicles, V2X, Vehicular Networks, and Virtualization. Each of these alludes to the main research question of this work: Can a vehicle accurately estimate traffic parameters using onboard resources shared through CV technology in a lightweight manner without utilizing centralized or roadside infrastructure? The goal of this work is to answer this single research question.

1.5 ROADMAP

In Chapter 2 we provide a literature review of traffic state estimation strategies. Next in Chapter 3, we discuss preliminaries of the Mobile Observer method including pass detection and MO method message protocols. In Chapter 4 the MO1 method is described and the concept of the tally is introduced. In Chapter 5, the concept of betweeness is introduced and the MO2 method is described as using codirectional vehicles. In Chapter 6 the MO3 method is discussed, which is similar to MO2, but involves oncoming traffic. Additionally, the MO3-Flow method is also discusses which mitigates some issues with the MO3 method. The MO1 and MO2 methods both utilize only V2V communications to aggregate the results, but the MO3-Flow method requires infrastructure to aggregate the results. In Chapter 7 the VRSU architecture is introduced which describes an architecture for running an RSU virtual using vehicles. Finally in Chapter 8, the conclusions of this work are provided along with limitations and ideas for future work.

CHAPTER 2

RELATED WORKS

The term *traffic state estimation* refers to measuring or inferring values of the key traffic state variables, such as density, flow, velocity, and delay by using a combination of observed and derived traffic-related data [16, 71]. Given that estimating traffic state is a fundamental task, it is not surprising that, over the past decades, hundreds of papers were written on this topic. The main goal of this chapter is to offer a review of recent work in this area.

By and large, traffic data is collected by sensors, either stationary or mobile. Legacy technologies for traffic state estimation that are still in widespread use today include inductive loop detectors (ILDs), video detection systems, infra-red tracking systems, acoustic tracking systems, and microwave radar sensors. By far the most prevalent among these devices are the ILDs embedded in the roadways every mile or fraction of a mile.

In due time it was noticed that vehicle-based measurements are an important source of input data for traffic state estimation that can supplement measurements from conventional stationary detectors such as ILDs. Loosely speaking, a probe vehicle is a specially-equipped vehicle in the traffic flow that acts as a moving sensor, collecting data and communicating the collected data to a roadside device (e.g. traffic light controller). Probe vehicles may or may not act in concert with other probe vehicles or with legacy stationary detectors.

The SAE J2735 standard was developed to meet the needs and requirements of various applications that depend on transferring, using V2X, information between vehicles and between vehicles and roadside infrastructure. One of the important benefits of the standard is to promote and support inter-operability among V2X applications through the use of standardized message sets, data frames, and data elements [72]. While probe vehicles have provided a more efficient and more cost-effective way of collecting real-time traffic data, their usefulness was limited by the fact that the probe vehicles were collecting data in isolation from each other. With the advent of vehicular networking, the ITS community has started to realize the benefits of networking together probe vehicles and also probe vehicles and roadside assets.

One of the valid concerns that the transportation community has looked into is whether or not the on-board sensors in probe vehicles can provide data that is sufficiently accurate for traffic state estimation purposes. Three early studies by Herrera *et al.* [73], Tao *et al.*

[74], and Uno *et al.* [75] have reported that on-board GPS devices, as well as cell phones, provide data that is highly accurate and can be used to estimate traffic state parameters.

Due to recent advances in vehicular networking and sensor technology and by exploiting existing and emerging computer and communications paradigms, the task of estimating traffic state parameters in *real-time* has become technically feasible, but remains a very challenging task [16].

This chapter is split into two sections. Section 2.1 reviews traffic state estimate strategies for traffic signal optimization. Then in Section 2.2 we review more general traffic state estimate strategies.

2.1 ESTIMATION FOR TRAFFIC SIGNAL OPTIMIZATION

2.1.1 BASICS OF TRAFFIC SIGNAL OPTIMIZATION

The reason traffic signals exist is to assign the right-of-way at intersections. Control of a particular direction is partitioned into three traffic light phases: green, yellow, and red. The interval from when the green phase begins and the red phase ends is the cycle time. The phase and cycle timings for each direction are controlled by a traffic signal controller.

Most traffic signals in the US run a set of predefined timing plans that set the signal's cycle length and green phase length based on the time of the day. In most cases, the optimization of the signal systems currently occurs off-line at either the isolated intersection or corridor level. One of the major disadvantages of this approach is that it requires data on traffic-turning movements be regularly collected to develop optimized traffic signal plans off-line. A second major disadvantage is that the time-of-day based signal timings do not adapt well to unexpected changes in traffic demand. For example, if an incident on the roadway network causes travel patterns to change significantly, the signals often cannot fully accommodate the changes in flow, resulting in possible traffic buildup and congestion. In order to ensure that the signals function as well as possible, they have to be retimed regularly to reflect current conditions. A retiming of the traffic signal is suggested at least every three years [76]. Unfortunately, due to budget or manpower limitations, transportation agencies often neglect to retime signals resulting in unnecessary delays to the traveling public.

Traffic signals can be a source of significant delays if cycle and phase lengths are not suitable for current traffic conditions. Under current practice, the process of developing optimal signal timing plans is resource-intensive. First, technicians must go into the field to manually collect vehicle and pedestrian volumes, types, and velocities during normal

usage and peak usage times. Additionally collision records are collected and analyzed to determine if varying the signal can make the intersection safer [76]. Then, this data is input into commercial signal optimization software packages such as Synchro or CORSIM [77] to develop optimal timing plans offline.

One solution that has been proposed is the use of adaptive traffic signal control (ATSC). ATSC systems use extensive detection to dynamically optimize flow along a corridor. These systems often use no fixed cycle of phase lengths and retime signals continually based on observed traffic flow [78]. There is often a local optimization that occurs to minimize delays at an individual intersection and a secondary global optimization that occurs along a series of signals on an arterial [79]. While some ATSC systems have been in existence for over two decades, they have not been adopted on a wide scale. More cost-effective systems have been developed with good results on corridor-level deployments [77]. However, due to their complexity, they have not been deployed beyond the corridor level [80].

2.1.2 STRATEGIES WITHOUT VEHICULAR INVOLVEMENT

The strategies discussed in this section are those which employ additional infrastructure to detect vehicles on the road. This new information is used by the traffic signal controller to make better decisions in controlling the traffic signals. Note that these strategies involve no contributions from the vehicles, other than their participation in the roadway traffic; the vehicles are merely bystanders being counted by a separate system.

The most basic strategy is the actuated traffic signal. A sensor is embedded in the road surface to indicate to the traffic signal controller when a vehicle is at the intersection. The traffic signal controller will make a decision to give green time to that direction. Actuated traffic signals come in three forms of control, Semi-Actuated, Full-Actuated, and Volume-Density. Semi-Actuated is where the sensors are only on the direction of the side streets. The side streets default to red, and only turn green when a vehicle is detected by the sensor. Full-Actuated is where all directions have sensors. The traffic signal controller makes decisions based on the data supplied by the sensors. Volume-Density is similar to Full-Actuated, but it uses more information from the sensors to make its decision. For example, the traffic signal controller may choose to remain green until a variable minimum gap between vehicles is achieved [81].

In the above scenarios, the most common sensor in use today is the ILD. Also in use are microloops, magnetic detectors, ultrasonic sensors, and radar. Some systems work by signaling a simple on or off. Other systems, as employed by the Volume-Density scenario

above, require more sophistication. By recording the times of when the sensors are on and off, the distance between vehicles can be determined as well as other details [19].

The basis for retiming a traffic signal comes from the knowledge of the traffic on the road. Roess *et al.* [19] mention three ways this is being done today. The first is by virtual detectors. A virtual detector is a camera system with specialized software able to count vehicles based on image processing technologies. Using this approach, vehicle count, velocity, and density, as well as queue length of the vehicles can be determined. Second, microwave detectors are used to determine when a vehicle passes a particular point on the road. Third are wireless detectors that behave much like ILDs, except the information is communicated to a controller wirelessly. One interesting approach of wireless detectors is proposed by Kwong *et al.* [82]; they utilize the magnetic detectors previously mentioned. Seven sensors are embedded within each lane a foot apart, perpendicular to the direction of the road. The sensors communicate wirelessly to an access point nearby. A similar set of sensors is located at a neighboring intersection on the same stretch of road. The system will record the magnetic signature and a timestamp for each vehicle as it passes over the sensors. As the vehicle passes over the other set of these sensors on the same stretch of road, the magnetic signature is matched and the timestamp is used to determine the velocity of the vehicle between the two points. Such a system is known as vehicle re-identification. The traffic information collected by such a system can be used by traffic signal controllers to make better decisions. Each of these strategies is based on using infrastructure to gather information about the vehicles, but none yet allows vehicles to participate in the retiming of traffic signals. The next section, Section 2.1.3, considers strategies that utilize the vehicles to supply information to the roadside infrastructure to aid in the collection of vehicle data.

2.1.3 STRATEGIES WHERE VEHICLES ARE PASSIVE PARTICIPANTS

The strategies in this and the next section that use VANETs use the following typical architecture, which we refer to as the VANET-based approach: a Vehicle Agent (VA) transmits important vehicle data to an Intersection Agent (IA). This IA is connected directly to the traffic signal controller. The IA/traffic signal controller aggregates the vehicle data to make a decision to change the traffic signal timing.

Kari *et al.* [83] includes some important system bookkeeping that other papers do not mention; particularly concerning keeping track of the status of the vehicle in the intersection. When a vehicle is within communication range of the IA it will first *check in* if it has not done so already. If the vehicle is leaving the intersection, it will *check out* if not already

done so. The check-in is done to inform the traffic signal controller that the vehicle is new and should be added to the vehicle list. The check-out is done to inform the traffic signal controller to remove the vehicle from the vehicle list.

An alternate approach to wireless transceivers is given by Bhuvaneswari *et al.* [84]. Here, the VA is a passive Radio Frequency Identification (RFID) tag and the IA is an RFID reader. Such an approach places limits on the solution. Specific limitations include the range of the communication, communication can only be one way, and typically only an ID can be stored on a passive RFID tag. Interestingly, Bhuvaneswari *et al.* [84] states vehicle information such as velocity is stored in a remote database. No details are given into how this information is stored into the remote database.

In the following sections, wireless transceivers utilizing V2X will be assumed as the method of communication between VA and IA. The next subsections investigate further differences in the VANET-based approach over the traditional approaches.

Advantages of VANET-based approach

A VANET-based approach offers several advantages over the infrastructure-based approaches from Section 2.1.2. The wireless transceiver allows the vehicle information to be collected anywhere within the transmission range of the IA. This allows the traffic signal controller to have a count of all vehicles over a large area, instead of one single spot on the road. A second advantage is that the vehicle can now give the IA additional information to enable the traffic signal controller to make more informed decisions. In most systems surveyed the VA will transmit the location of the vehicle. With this information the traffic signal controller can plot the vehicle on the road, instead of just knowing the number of vehicles on the road. Some systems surveyed [85, 86, 87] also require vehicles to send their velocity. This provides the traffic signal controller with a rough estimate of when the vehicle will reach the intersection, and whether it will reach a green phase. In other systems the vehicles may send information identifying it as an emergency vehicle, the type of vehicle, or expected carbon emissions depending on what the optimization algorithm requires. The IA may also send information to the vehicle, an option the traditional infrastructure-based approaches do not allow. In the strategy of Kari *et al.* [83] the intersection geometry is sent to the vehicle. It is assumed that this additional information is used by the vehicle to aid in calculating the expected carbon emissions. Finally, the IA may require a small amount of computation to be handled by the vehicle. The strategy of Kari *et al.* [83] requires the vehicle to calculate the expected amount of carbon emissions. Also, some papers ask the

vehicle to determine its own expected time of arrival [83, 86]. This alleviates some computing needs from the traffic signal controller. The computation required by the systems in this section is limited; however, utilizing vehicles to perform computations could lead to interesting new approaches in the future.

Disadvantages of VANET-based approach

In most papers surveyed that utilize the VANET-based approach, each vehicle is assumed to have a wireless transceiver. The systems would degrade if non-enabled vehicles were considered in the traffic. Few papers surveyed suggested any approaches to handle a vehicle without a wireless transceiver, or a vehicle with a malfunctioning wireless transceiver.

A particularity concerning disadvantage that is not discussed in detail in the literature is that traditional ILDs offer a glimpse of traffic at a single point and, consequently, it can determine a count of traffic per lane. A vehicle attempting to locate itself on the roadway is unable to determine its location with this accuracy. Kari *et al.* [83] and Wenjie *et al.* [85] suggest that the lane can be determined by GPS or by triangulation, but evidence is not provided that supports these claims. These strategies rely on knowing the exact lane; the effectiveness of each is degraded if the vehicle's lane is approximated.

The final disadvantage is the need for more computing power from the traffic signal controller. Current traffic signal controllers make decisions based on fixed timings or from inputs from a small number of sensor inputs. Many of the algorithms suggested in this section are only slightly more complex; however, there is a category explained later in Section 2.1.3 which requires a huge amount of processing power. The computing power required of the traffic signal controller must be considered.

Reactive versus predictive traffic signal optimization

One way to handle traffic on the roads is to first determine the traffic at intersections, and then retime the traffic signal based on these conditions. Such a system is said to be reactive. In the infrastructure-based strategies, most systems are reactive because they adjust phase and cycle lengths based on the number of vehicles waiting at the intersection. VANET-based approaches offer a bit of intelligence over purely reactive systems. Many require the VA to send the expected time of arrival (at the intersection), or information to determine location and velocity. With this information, the traffic signal controller can determine when each vehicle will arrive at the intersection. The algorithms use this information to reduce times

waiting at the intersection. In these cases, the level of prediction is determined by the range of the IA's wireless signal [83, 84, 85, 86, 88, 89].

McKenney *et al.* [90] proposed a system that offloads prediction to the location of the previous intersection by allowing communication between intersections. The data for vehicles passing through an intersection is communicated to the neighboring intersection along the vehicle's route. In this approach, the data collected at each intersection is shared with neighboring intersections to increase the level of prediction.

In Predictive Microscopic Simulation Algorithm (PMSA) [87], each of the vehicles on the road communicate their location, heading, and velocity to the traffic signal controller. Those within 300 meters of the intersection are mapped into a microscopic simulation to determine the whereabouts of each vehicle after a horizon of 15 seconds. The algorithm bases whether the vehicle will turn or go straight on the lane they are in. For example, a vehicle in the turn lane will likely turn, a vehicle in the straight lane will likely not turn. For lanes that may go straight or turn, an equal ratio is used to make a prediction. After 15 seconds another simulation is run to again predict the vehicle movements. In this algorithm, the traffic signals are retimed based on the vehicle's predicted location.

Prediction is an interesting topic in its own right; however, the predictions are based at least partially on current estimates for traffic parameters. This thesis will not consider prediction.

Isolated versus coordinated traffic signal optimization

Many of the papers surveyed consider traffic signal optimization in the context of an isolated intersection. By extending the algorithm to coordinate on a city-wide network, the "perfect" traffic timing can be determined for the entire network. Finding this "perfect" timing involves a complex calculation that requires an amount of computing power that is unattainable under current practice. Cheng *et al.* [91] proposed a network-wide algorithm; however, because of its complexity several shortcuts were taken. McKenney *et al.* [90] proposed that vehicle information be shared with its neighbors; however, it does not attempt to coordinate signal timings with neighbors. Most of the surveyed literature offer strategies in isolation from other signals [83, 84, 85, 86, 87, 88, 92]. These algorithms work in isolation, but still manage to get better results than traditional fixed-time signals.

2.1.4 STRATEGIES WHERE VEHICLES ARE ACTIVE PARTICIPANTS

In the previous section, some systems offloaded certain calculations to the vehicles on the

road. These calculations were limited to simple ones, e.g. estimation of arrival time to the next intersection and estimate of carbon emissions. In this section, we present approaches where the vehicles are responsible for more complex calculations.

Vehicles on the roadway are being outfitted with more and more types of sensors, as well as more powerful computers to aggregate the data collected by these sensors. Additionally, with the advent of initiatives such as US Department of Transportation Connected Vehicle Research [93] vehicles of the future will communicate with each other and with the roadside infrastructure. In the previous sections, we covered strategies where the vehicle is expected to supply information about its location or its velocity to help the traffic signal controller make a decision. In this section we explore alternative approaches to enlisting the vehicles computational capabilities. We believe the future of traffic signal optimization lies in offloading more of the computing process to the vehicles on the road.

Maslekar *et al.* proposed CATS [94], a system that works in a way similar to those discussed in the previous section; however, the density of vehicles on each link is determined by the vehicles running a clustering algorithm. Clusters are formed based on the direction the vehicle will take at the upcoming intersection. The cluster head for each cluster will communicate to the traffic signal controller the density of the cluster, and the length of the cluster. The vehicle will exit the cluster once it leaves the intersection. The interesting work here is that the vehicles themselves determine the number of vehicles on the road.

Xiang-ya *et al.* [95] proposed a similar approach where the vehicles are responsible for counting the vehicles on the road. They offer a traffic information system that can be used to report the vehicles on a particular stretch of road. The network of roads is separated into regions by a virtual grid. Streets are split into segments and each segment is given a level based on the type of road; expressway, main arterial, minor arterial, collector, or local street. A cluster head can only be chosen from certain types of street segments defined as valid. For example, a vehicle on an expressway may be a poor candidate for cluster head since it will not be in the cluster for a long time. The vehicle closest to the center of the grid, and on a valid street segment, is chosen as cluster head and acts as a location server for a determined duration. Each vehicle on the road will update its local location servers with its ID, velocity, direction, and location upon traveling a set distance threshold. Local location servers are those within an $R \times R$ square; all others are remote location servers. Location servers communicate with neighboring location servers to send updates on vehicle locations. The location servers closest to a vehicle will have the most up-to-date information on the location of that vehicle.

Since each vehicle on the road is responsible to update its location server, these location servers will contain all the vehicle locations within its region of the grid. Each location server will aggregate vehicle information into a data structure that contains the street ID, segment ID, and list of vehicles.

2.1.5 ESTIMATING QUEUE LENGTH AND DELAY AT SIGNALIZED INTERSECTIONS

Estimating queue length and delay at signalized intersections has received a lot of attention in the literature and numerous probe vehicle-based approaches to estimate queue length have been reported in the past decade. For example, Comert and Cetin [96] use probe vehicle information to develop accurate estimates of queue lengths and delay at signalized intersections. Real-time information about queue lengths and delay enable optimal control of the available capacity. This work is continued by Comert [97] where simple analytical models are developed for estimating the queue length from probe vehicle information collected at traffic lights.

Li *et al.* [98] have estimated queue length under CV technology using probe vehicles, ILDs, and fused data. Cheng *et al.* [99] have looked at the problem of signalized intersection management. Specifically, they developed a methodology aimed at estimating queue length as a performance indicator. Their method can provide cycle-by-cycle queue length estimates using sample probe vehicle trajectory data collected from NGSIM traffic traces, synthetic data, and GPS data sets collected by the authors. Shladover and Li [100] have investigated probe vehicle sampling strategies for improving MOE relevant to traffic signal control. Cai *et al.* [101] have developed a methodology to estimate in real-time cycle-by-cycle queue length at signalized intersections by utilizing the data collected from an upstream point sensors and traveling trajectory of probe vehicles. Hao and Bai [102] have used probe vehicles to estimate travel time on arterials.

Very recently, Zheng and Liu [103] employed two connected probe vehicles to determine the number of vehicles in a queue. Their method uses the stopping position of the lead probe vehicle to determine the number of queued vehicles in front. The authors show how the trailing probe vehicle can either estimate the number of vehicles in front, or estimate an upper bound on the number of vehicles between it and the lead MO.

There are several interesting research challenges related to the efficient use of probe vehicles as moving sensors. The first such challenge is to determine the penetration rate of probe vehicles that can guarantee a certain quality of traffic state estimation [11, 104, 105].

The second interesting question concerns sampling and signaling. As already mentioned, the SAE J2735 standard has prescribed sampling standards. However, there is still a good deal of work investigating various other sampling strategies [100]. To reduce the number of messages that must be sent to the centralized server, Fusco *et al.* [106] uses a fixed interval to sample probe vehicles. Lim *et al.* [107] uses adaptive sampling to reduce the number of transmissions.

2.2 TRAFFIC STATE ESTIMATION FOR GENERAL CASES

2.2.1 ESTIMATING TRAFFIC STATE PARAMETERS USING PROBE VEHICLES AND HYBRID APPROACHES

Popular approaches to estimating traffic state parameters involve a combination of legacy and mobile sensors. In these approaches, probe vehicles play the role of mobile sensors while the legacy equipment helps fuse the data. Initially, the probe vehicles were not networked together. Instead, they would send the collected data to various aggregation points.

In an early paper, Nanthawichit *et al.* [12] proposed a hybrid method for dealing with data collected from probe vehicles along with conventional ILD data to estimate the state of traffic. They integrated probe data into a Kalman filter in which the state equations are represented by the macroscopic traffic flow model. As it turns out, Kalman filter technology has been used extensively in hybrid approaches. Wang *et al.* [15] used a hybrid approach to estimate the state of freeway traffic based on an extended Kalman filter technique. Quite recently, Wang *et al.* [108] used a hybrid strategy involving genetic algorithms and Kalman filter techniques to estimate highway traffic parameters.

Qiu *et al.* [109] proposed a strategy for estimating freeway traffic density using ILD data aggregated with probe vehicle data. They showed that their hybrid approach offers a substantial increase in the accuracy of the estimation. Similarly, Van Lint and Hoogendoorn [110] showed how to obtain robust estimates of traffic state parameters by fusing data from probe vehicles and various other sensors. Tyagi *et al.* [111] have devised an approach that involves harvesting acoustic data collected by microphones in probe vehicles and using it to acoustically classify the traffic. Using a Bayes classifier they estimated the traffic density. Anand *et al.* [112] used a combination of digital videos taken by roadside cameras and probe vehicles with on-board GPS devices to collect and process traffic flow and density data. Zhang *et al.* [113] have studied the accuracy of highway velocity estimation using a combination of loop detectors and probe vehicles. Similarly, Ambuhl *et al.* [114] proposed

a data fusion algorithm using both ILD and probe vehicle data. They showed that their hybrid approach offers an estimate better than that either can individually provide. More recently, Fountoulakis *et al.* [115] have studied the accuracy of highway velocity estimation using a combination of loop detectors and probe vehicles.

The Mobile Millennium project at UC Berkeley uses the GPS in the cellular phones in (probe) vehicles to infer information about traffic state [79, 116]. Herrera *et al.* [73] used GPS-enabled smartphones to estimate traffic parameters. They showed that relying on traffic data collected by cell phones seems to work best in environments that experience a high concentration of vehicles and less well on less well traveled highways where there is no "critical mass" of cell phones or probe vehicles. Also, it was noted that GPS devices are power-hungry and are afflicted with reception problems in urban canyons with tall buildings. In addition, GPS-related privacy concerns are significant enough to prevent the widespread adoption of such tracking technologies as a source of reliable information for monitoring traffic conditions on large-scale urban arterials.

Finally, it is worth mentioning that all smartphones-based systems rely on the existence of an operational mobile phone system. As a result, these systems fail to work in the case of emergencies, where some of the infrastructure is no longer operational. Such is the case, for example, in evacuation scenarios in the wake of a hurricane, terrorist attack, earthquakes, and the like, when part of the installed infrastructure is temporarily out of commission. By contrast, V2V and some V2I-based radios do not rely on operational infrastructure and are useful in all sorts of emergencies [117]. With this in mind, Yan *et al.* [24] have proposed NOTICE, a secure and privacy-aware system for the automatic detection of traffic parameters and for the dissemination thought V2I of related traffic advisories. NOTICE uses belts of piezoelectric elements embedded in the roadway to detect variations in average velocity, traffic density, and traffic flow. NOTICE supplements this information by collecting data from passing vehicles.

2.2.2 CONNECTED VEHICLES-BASED APPROACHES

Argote-Cabanero *et al.* [11] use CV technology to estimate, in real-time, various MOE for signalized arterial traffic. They make the point that CV is a promising mobile data source that can provide real-time information useful for evaluating traffic conditions. They are especially interested in average velocity, number of stops, acceleration noise and delay. The main contribution of the paper is to develop a methodology to determine the minimum market penetration rate of CV to guarantee accurate MOE estimates as a function of traffic

conditions, signal settings, sampling duration, and MOE variability.

Quite recently, Yu *et al.* [118] have reviewed several approaches for managing intersection traffic using a combination of CV and automated vehicle technology. Bekiaris-Liberis *et al.* [119] propose estimations of highway traffic state by using a combination of CV and conventional vehicles. They use the average velocity of connected vehicles and assume that the conventional vehicles have the same average velocity.

Khan *et al.* [10] proposed to enhance the CV approach by borrowing data fusion techniques from artificial intelligence (AI). In addition, they use pre-deployed roadside units for data collection.

More recently, Grumert *et al.* [71] proposed a CV-based traffic density estimation method using a hybrid approach that requires infrastructure equipment. Their method uses connected vehicles to report their positions including information about the current location, direction, and velocity. In addition, they need stationary detectors to count and report the total number of vehicles passing the detector in a given time interval.

Some researchers have used headway distance measurements to estimate traffic state, as did Yan and Olariu [120]. Interestingly, Zheng *et al.* [121] use headway distances from probe vehicles and a stochastically measured headway distance from velocities of non-equipped vehicles to estimate traffic state based on the Newell-Franklin velocity-spacing relation.

Given the importance of the Fundamental Diagram (FD) to a roadway, some researchers directly estimate it. Seo *et al.* [105] uses a statistical estimation algorithm for a triangular FD using probe vehicles. The parameters of the FD are determined using the time two probe vehicles spend in an area as well as length of the area. No additional information from non-probe vehicles is necessary. Using several of these readings, the backward wave velocity can be estimated as well as the free flow velocity. Using this method, authors suggest the FD can be estimated within a few months.

van Erp *et al.* [122] propose a method using mobile observers traveling with traffic and in oncoming traffic to observe relative flow. The paths of the vehicles form an enclosed area from which Edie's generic definitions for flow and density can be used. The authors give a method for estimating a triangular FD given several estimates for q and k. van Erp *et al.* [14] proposes a similar method to estimate traffic density.

CHAPTER 3

PRELIMINARIES TO MOBILE OBSERVER METHODS

This chapter starts with an explanation of the Moving Observer Method that was proposed by Wardrop and Charlesworth in 1954.

This is followed by two preliminary technologies required for the Mobile Observer methods that will be presented in later chapters. First, it is important in the methods for an enabled vehicle to identify the moment in time it passes another vehicle. Methods of pass detection are presented in Section 3.2. Then in Section 3.3, the message formats and protocols for each of the messages sent via V2V and V2I are presented in detail.

3.1 MOVING OBSERVER METHOD

In 1954, Wardrop and Charlesworth [26] proposed a moving observer method for estimating traffic flow. A test vehicle will drive the length of the test area with the traffic to be measured. While driving with the traffic, a tally of the number of times the test vehicle passes another vehicle minus the number of times the test vehicle is passed by another vehicle is maintained by a passenger in the vehicle. Additionally, the time spent driving the length of the test area with traffic and the average velocity of the test vehicle is recorded.

The test vehicle will then turn around and keep a tally of the oncoming traffic. When driving against traffic, the tally is the number of vehicles in the oncoming traffic the test vehicle meets. The time spent in the test area and the average velocity of the test vehicle is recorded.

By driving both with and against traffic, the observers are able to collect enough information to determine the flow of vehicles on the roadway in the direction of interest. We now explain their methodology and proof.

Wardrop and Charlesworth are interested in the flow of traffic for a stretch of roadway of length l. Here they assume there is a constant flow of vehicles. In their proof, they split the traffic into several sub-flows, where a generic sub-flow i will have a velocity v_i, a flow q_i, and each vehicle in the sub-flow will spend time t_i in the test area. Note that the flow of traffic is the summation all the sub-flows, $q = q_1 + \ldots + q_i + \ldots + q_n$.

When driving with traffic the test vehicle will have a velocity v_w and will spend a time t_w in the test area. For a given sub-flow, the test vehicle will have a *tally* of τ_i. This tally

is the number of times a vehicle in the sub-flow i passes it minus the number of times the test vehicle passes another vehicle.

The relative flow the vehicle will observe while driving with the traffic is:

$$q_{w_i} = \frac{q_i(v_i - v_w)}{v_i} \tag{7}$$

The tally, τ_i for a generic sub-flow will be:

$$\tau_i = q_i(t_w + t_i) \tag{8}$$

Now, by adding all the sub-flows from Equation (8) we get:

$$\tau = q(t_w + t_{avg}), \tag{9}$$

where t_{avg} is the average time a vehicle spends traveling the test area. We note that there are two unknowns in Equation (8), namely q and t_{avg}. In the method, the test vehicle will travel in the opposite direction to get a second equation to solve for q.

Similarly, when driving against traffic the test vehicle will have a velocity v_a and spend a time t_a in the test area. For a given sub-flow, the test vehicle will have a *tally* of τ_i'. This tally is the number of times a vehicle in the sub-flow i the test vehicle meets. Note that the test vehicle is in the opposite moving traffic than the vehicles in the sub-flow.

The relative flow the vehicle will observe while driving against the traffic is:

$$q_{a_i} = \frac{q_i(v_i - v_a)}{v_i} \tag{10}$$

The tally, τ_i, for a generic sub-flow will be:

$$\tau_i' = q_i(t_a + t_i) \tag{11}$$

Now, by adding all the sub-flows from Equation (11) the result is:

$$\tau' = q(t_a + t_{avg}), \tag{12}$$

where t_{avg} is the average time a vehicle spends traveling the test area. Finally, the expression for flow, q, is determined by adding both Equations (9) and (12) and solving for q:

$$q = \frac{\tau + \tau'}{t_a + t_w} \tag{13}$$

As mentioned in Section 1.2.5, there are several issues with this moving observer method that must be addressed.

First, to get results comparable to the Stationary Observer method, the test vehicle must perform multiple runs through the traffic in both directions, during which the flow of vehicles in practice will not remain constant. Additionally, this method requires the test vehicle to be able to observe oncoming traffic. It is not guaranteed that the test vehicle will be able to observe the oncoming traffic.

Starting in Chapter 4, we propose a modernized variants of the Moving Observer method that we call the Mobile Observer Methods. Each are based on a very same idea of the *tally* as proposed by Wardrop and Charlesworth; however, the methods are utilize the capabilities of modern day vehicles.

3.2 PASS DETECTION

In this section methods for approximating the meeting time of two vehicles are presented. In Subsection 3.2.1 we offer the details of a method to approximate the meeting time that requires only V2V radio communications. Next, in Subsection 3.2.2 we outline a method that assumes that one of the vehicles has on-board radar.

3.2.1 PASS DETECTION FOR USING V2V ONLY

As will be discussed in Chapters 4, 5, and 6 some of the Mobile Observer Methods require vehicles to communicate and know the exact moment two enabled vehicles pass in codirectional traffic or meet in oncoming traffic.

Consider vehicles X and Y, traveling in the opposite directions along an urban arterial corridor. We allow either of X and Y to stop, for various reasons, as long as they finally meet.

We assume that V2V communications conform to the SAE J2735 standard using their CV2X-compliant radios. As introduced in Section 1.2.11 the standard prescribes that the vehicles transmit Basic Safety Messages every δ time units. As per the standard δ is 0.1 seconds; however, there is debate whether this is too often, especially when considering a congested network. With this in mind, X and Y will both transmit messages. We note that each vehicle will not transmit its BSM at the same time; however, it should be sufficiently close for this method.

As per SAE J2735, the Basic Safety Message includes the following three key fields for estimating the time two vehicles will pass:

- latitude and longitude,
- velocity and heading, and
- a temporary ID.

Using the position data from the BSM and its digital map, the vehicle, by means of a map matching algorithm, will determine the location of the vehicle on a digital map. Additionally, using the velocity and heading from the message, the vehicle can be determined to be in oncoming or codirectional traffic. Using several BSM messages from the same Temporary Id, over time the vehicle can be tracked.

This is illustrated using the following example. Consider two vehicle X and Y within communication range of one another. Vehicle Y will be the host vehicle and will receive a BSM from vehicle X. The same method will also be performed by vehicle X upon receiving vehicle Y's BSM.

Upon receiving the BSM from vehicle X, vehicle Y will use a map matching algorithm to place X on a digital map; it will then determine if both vehicles are on the same road. If they are on the same road, the host vehicle Y will attempt to estimate the time they will pass. To do this, the distance between the two vehicles is important to determine. More specifically, the roadway distance between the two vehicles must be determined. This is done by transposing both vehicle's locations onto a one dimensional road. The location of vehicle X at time t is represented by $x(t)$ and the location of vehicle Y at time t is represented by $y(t)$.

Next, vehicle Y will determine if vehicle X is codirectional or is in oncoming traffic. The velocity and heading from the BSM form a vector representing the vehicle X's velocity at time t, $v_X(t)$. Based on the heading, the vehicle is determined to be in codirectional or oncoming traffic. Specifically, if both vehicles share the same heading, they are considering codirectional; otherwise, they are in oncoming traffic. If vehicle X is determined to be in codirectional traffic, its velocity at time t will be translated as a positive value. Otherwise, if vehicle X is determined to be in oncoming traffic, its velocity at time t will be translated as a negative value.

All the information needed to estimate the time the two vehicles will pass has been determined. Vehicle Y will produce an estimate θ_i of the time the two vehicles will pass:

$$\theta_t = \frac{y(t) - x(t)}{v_X(t) - v_Y(t)} \tag{14}$$

This basic equation can be used to estimate the time the two vehicles will meet in oncoming traffic or pass in codirectional traffic.

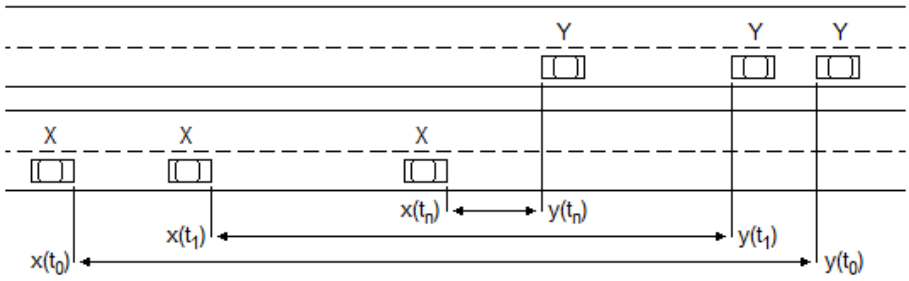


Fig. 6: Estimating the meeting time between vehicles X and Y running in opposite directions.

We first consider vehicle X and vehicle Y traveling in opposite directions on the same road; see Figure 6. At time t_0 vehicle X will transmit a BSM. Vehicle Y will receive the message. Based on the position, heading, and velocity, vehicle Y will determine that vehicle X is in oncoming traffic, its position $x(t_0)$, and its velocity $x_Y(t_0)$. Vehicle Y will then use Equation (14) to estimate θ_0 the time the vehicles will meet. Since $\theta_0 > t_0 + \delta$, the two vehicles are not expected to meet before the next BSM is sent by vehicle X. This will continue until time t_n when vehicle X sends another BSM. Again, the same process will occur resulting in vehicle Y determine a new time to meet, θ_{t_n}. In this case $\theta_{t_n} > t_n + \delta$, so the two vehicles are expected to meet before the next BSM. Vehicle Y will store this time as the moment in time both vehicles met. This time to meet is important for each of the MO3 methods as discussed in Chapter 6.

We now consider codirectional vehicles; see Figure 7. In this case, the velocities for both vehicles will be positive. At time t_0, vehicle X will transmit a BSM. Y will receive the message and will determine the following three key details using the position, velocity, and heading from the BSM:

- if X is codirectional or not,
- X's position $x(t_0)$, and

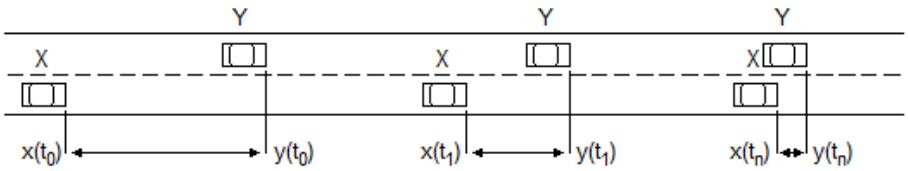


Fig. 7: Estimating the meeting time between vehicles X and Y running in the same directions.

- velocity $x_Y(t_0)$.

Vehicle Y will then use Equation (14) to estimate θ_0, the time the vehicles will pass. Since $\theta_0 > t_0 + \delta$, the two vehicles are not expected to pass before the next BSM is sent by X. This process will continue until t_n, when $\theta_{t_n} > t_n + \delta$. This means X and Y are expected to pass prior to the next BSM message. Vehicle Y will store θ_{t_n} as the of passing. This time is important for MO1, MO2, and both MO3 methods.

3.2.2 ESTIMATING THE MEETING TIME WHEN ON-BOARD RADAR IS AVAILABLE

We now outline an alternate approach where vehicle Y has on-board long-range radar, or other method of tracking vehicles like lidar or virtual sensors with a range equivalent to 150 meters. For purposes of illustration, we will assume radar. Additionally vehicle X is not enabled.

In this case, referring to Figure 8, vehicle Y knows its own location $y(t)$, its own velocity $v_Y(t)$ and, by using its on-board radar, it can approximate:

- the velocity $v_X(t)$ of X at time t,
- the angle α of arrival of the reflected wave from X to back to Y, and
- the distance $d(X, Y, t)$ between X and Y at time t.

Now, using elementary algebra, the meeting time, t_2, can be approximated as

$$t_2 \approx t_1 + \frac{d(X, Y, t_1) \cdot \cos\alpha}{v_X(t_1) + v_Y(t_1)}. \tag{15}$$

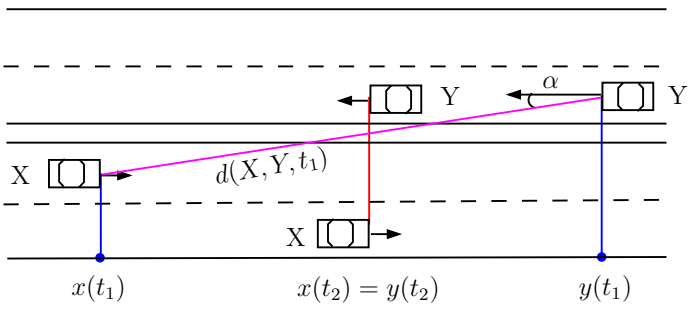


Fig. 8: Illustrating an alternate method for estimating t_2.

A third possible method for approximating the meeting time is for Y to continuously sample the angle of arrival α until it is approximately $\frac{\pi}{2}$.

3.3 MO METHOD MESSAGE PROTOCOLS

We now offer the details of each of the MO Method messages that will be sent.

First, in Section 3.3.1 we discuss the Test Area Advertisement where the TCM will advertise the location and geometry of the areas it needs traffic parameters approximated. Then in Section 3.3.2 the Traffic Parameter Collection is described. Next in Section 3.3.3 and 3.3.4, the two messages making up the MO Partner Negotiation. These are the messages required to establish an MO Partner relationship. Then, the Tally Exchange Message is described in Section 3.3.5. Finally, the use of the BSM in MO is described in Section 3.3.6.

3.3.1 TEST AREA ADVERTISEMENT

The TMC must have a means of advertising which stretches of road need traffic parameter estimates. It may be that one day all stretches of road will require traffic parameter estimates; however, given the concern that bandwidth be conserved [123], we assume that test areas will be advertised and changed regularly. This will be done with the Test Area Advertisement (TAA).

This advertisement is a good candidate for the WSA and WSM messages. An RSU or a group of RSUs will each provide a service for advertising the local MO Test Areas. Each RSU will include in their WSA message the advertisement for the TAA service, which includes

the Provider Service IDentifier (PSID) representing the TAA service. Enabled vehicles will receive the WSA message and will subscribe to the service. Then the RSU will regularly send WSM messages with a list of Test Area Advertisements.

TABLE 4: The TAA Format

Field	required	size
Test Area ID	required	2 bytes
MO Method Type	required	2 bits
Geometry	required	variable bytes

The Payload of the WSM data field includes a list of TAAs with the following format that is also shown in Table 4:

Test Area ID - The Test Area ID is a unique identifier for the test area. This is used by other messages to identify the Test Area. It is 2 bytes long giving a total number of 65536 possible Ids.

MO Method Type - The MO Method Type field identifies which MO method that the vehicle should utilize. Each of the MO methods are used for certain circumstances. It is important to advertise which method will be used. This field is 2 bits, with the following assignments:

1. MO1,
2. MO2,
3. MO3, and
4. MO3 - Flow.

Geometry - The Geometry field contains subfields that give identifying information to the road as well as descriptions of the start and end boundaries of the test area. The description of the Geometry section will be described below. We note here that the minimum size of the Geometry field is 234 bits and the maximum size is 1,130 bits.

This completes the TAA message format.

The Geometry Field from the TAA message is now described in full detail, as well as in Table 5(a):

Road ID - The Road ID field is a 4 byte unique identifier that represents a road.

Heading - The Heading field is a 2 byte value that represents the direction of travel of the roadway from the Boundary Start line to the Boundary End Line. This field is 2 bits, with the following assignments:

1. North,
2. South,
3. East, and
4. West.

Boundary Start - The Boundary Start field represents a line that marks the starting line of the test area. It is represented with a Boundary Type format. The Boundary format will be described below. We note the minimum size of the boundary field is 100 bits, and the maximum size is 548 bits.

Boundary End - The Boundary End field represents a line that marks the end line of the test area. It is also represented with a Boundary Type format.

This completes the Geometry Field format.

The Boundary Field from the Geometry field is now described in full detail, as well as in Table 5(b):

Reference Point - The Reference Point represents a point that will be used as a reference point to draw a boundary line using multiple nodes. It uses the Position2D field type which includes the latitude and longitude values. The precision of each field is 1/8th micro degrees. Both fields together use 8 bytes.

Number of Nodes - The Number of Nodes field contains an integer value representing the number of nodes in the Node List field. This field has a size of 4 bits, which represents a maximum of 16 nodes in the node list. There must be at least 2 nodes in the list.

Node List - The Node List field contains a variable number of node fields. Each node field contains an x and y offset represented in centimeters from the Reference Point. Each node is ordered and are used to draw a line from one side of the road to the other designating either the start or end line of the boundary. The x and y offset fields are both each 2 bytes and represent a signed integer value between -32,768 and 32,767. There must be between

2 and 16 nodes in this list. This means the minimum size of the node list is 64 bits, or 8 bytes. Also, the maximum size of the node list of 512 bits, or 64 bytes.

This completes the Boundary Field format.

TABLE 5: Geometry and Boundary Format

(a) The Geometry Format

Field	required	size
Road ID	required	4 bytes
Heading	required	2 bits
Boundary Start	required	variable bytes
Boundary End	required	variable bytes

(b) The Boundary Format

Field	required	size
Reference Point	required	8 bytes
Number of Nodes	required	1 byte
Node List	required	variable bytes

3.3.2 TRAFFIC PARAMETER COLLECTION

In addition to the RSU advertising test areas and the MO method to use, the RSU must also have a means of retrieving density, flow, and tally information from the MO methods.

The Traffic Parameter Collection (TPD) message will use the WSA and WSM messages types. An RSU or a group of RSUs will each provide the service of collecting the results of the vehicles. The RSU or group of RSUs will advertise the TPC service by using a WSA. The WSA will include a Provider Service IDentification (PSID) that is unique to this service.

Once MO partners have an estimate for a traffic parameter, they will send an WSM with the fields, as described below. Then the RSU or any other entities subscribed to the PSID can listen and receive the message with the traffic parameter details.

The Payload of the WSM data field has the following format that is also shown in Table 6:

TABLE 6: The TPC Format

Field	required	size
Test Area ID	required	2 bytes
MO Method Type - Extended	required	3 bits
MO Measurement Data	required	variable bytes

Test Area ID - The Test Area ID is a unique identifier for the test area. This is used by other messages to identify the Test Area. It is 2 bytes long giving a total number of 65,536 possible Ids.

MO Method Type - Extended - The MO Method Type - Extended field is the combined bits of the 2 bit MO Method Type field and a single Summary bit. The MO Method Type is the same format and values as the MO Method Type Field from the TAA message. The Summary bit is used to specify the MO Measurement Data as being from a single vehicle or a summary of the vehicle data. The Summary Bit will be a 1 if it is a summary of other vehicles MO measurements and a 0 otherwise. This bit will be discussed more in Chapter 7.

MO Measurement Data - The MO Measurement Data is a variable format field that contains measurement data in a format that depends on the MO Method Type. Below, each of the fields are described; see Table 7, for the specific fields for each method.

This completes the TPC message format.

The MO Measurement Data field for the MO1 Measurement Type is described below, and also in Table 7(a):

Start Time - The Start Time field is the time the test vehicle passes into the MO1 test area. The Time fields use the DTime field from the DSRC specification [48] which contains the three fields: DHour, DMinute, and DSecond. DHour is a 1 byte field containing the hour from 0 to 23. DMinute is a 1 byte field containing the minute from 0 to 59. DSecond is a 2 byte field containing the microseonds from 0 to 60,000.

End Time - The End Time field is the time the test vehicle exits the MO1 test area. The End Time uses the same DTime field from the specification.

Density - The Density field is the estimated density as determined by the MO1 method.

TABLE 7: Format for the three types of MO Measurement Data

(a) Format for MO1

Field	required	size
Start Time	required	4 bytes
End Time	required	4 bytes
Density	required	2 bytes
Flow	required	2 bytes
Tally	required	1.5 bytes

(b) Format for MO2 and MO3

Field	required	size
Start Location	required	8 bytes
End Location	required	8 bytes
Time	required	4 bytes
Density	required	2 bytes

(c) Format for MO3-Flow

Field	required	size
Start Time	required	4 bytes
End Time	required	4 bytes
Tally/Flow	required	2 bytes

This will be included if the test vehicle has aggregated results locally; otherwise, it may be left as all zeros. The density represents the density in km; however, is multiplied by 100 before encoding. Being 2 bytes long it can represents a density between 0.00 to 665.35 with 2 digits of accuracy.

Flow - The Flow field is the estimated flow as determined by the MO1 method. This will be included if the test vehicle as aggregated results locally; otherwise, it may be left as all zeros. The Flow represents vehicles per hour and is multiplied by 10 before encoding. Being 2 bytes long, it can represent a flow between 0.0 and 6,653.5 with 1 digit of accuracy.

Tally - The Tally field is the tally as determined by the test vehicle. This is always sent; however, its purpose is to make the tally data available so the density, flow, and velocity can be aggregated at the RSU. The Tally field 1.5 bytes (12 bits) long; it can represent a value from 0 to 4,095.

This completes the Measurement Data format for MO1.

The MO Measurement Data field for the MO2 and MO3 Measurement Type is described below, and also in Table 7(b):

Start Location - The Start Location field is the location of the trailing vehicle representing

the first point in the density measurement. The location fields uses the Position2D field from the DSRC specification [48]. It includes the latitude and longitude values. The precision of each field is 1/8th micro degrees. Both fields together use 8 bytes.

End Location - The End Location field is the location of the leading vehicle representing the second point in the density measurement. The End Location uses the same Position2D field format as Start Location.

Time - The Time field is time of the density measurement. The Time fields use the DTime field from the DSRC specification [48] which contains the three fields: DHour, DMinute, and DSecond. DHour is a 1 byte field containing the hour from 0 to 23. DMinute is a 1 byte field containing the minute from 0 to 59. DSecond is a 2 byte field containing the microseonds from 0 to 60,000.

Density - The Density field is the estimated density as determined by the MO2 method. The density represents the density in km; however, is multiplied by 100 before encoding. Being 2 bytes long it can represents a density between 0.00 to 665.35 with 2 digits of accuracy.

This completes the Measurement Data format for MO2 and MO3.

The MO Measurement Data field for the MO3-Flow Measurement Type is described below, and also in Table 7(c):

Start Time - The Start Time field is the time the first of the MO partner passes into the test area. The Time fields use the DTime field from the DSRC specification [48] which contains the three fields: DHour, DMinute, and DSecond. DHour is a 1 byte field containing the hour from 0 to 23. DMinute is a 1 byte field containing the minute from 0 to 59. DSecond is a 2 byte field containing the microseonds from 0 to 60,000.

End Time - The End Time field is the time the second of the MO partner exits the test area. The End Time uses the same DTime field from the specification.

Tally/Flow - The Tally/Flow field is a dual purpose field. If the Summary Bit of the MO Method Type - Extended field is a 0, then this will be a Tally field. If the Summary Bit of the field is a 1, then this will be a Flow field. The purpose of this field is to send the tally to the RSU so it can aggregate to determine the flow. The Tally field is 1.5 bytes (12 bits) long; it can represent a value from 0 to 4,095. The Flow field is the flow that was aggregated by the RSU. It is multiplied by 10 before encoding. It is a 2 byte (16 bits) field that can represent a flow between 0.0 and 6,653.5 with 1 digit of accuracy.

This completes the Measurement Data format for MO3-Flow.

3.3.3 MO PARTNER REQUEST

The MO Partner Request (MPR) message is the first of two messages in the MO Partner Negotiation. The purpose of this message is for an enabled vehicle to advertise themselves as being enabled.

In the MO methods, with exception of MO1, it is important to know who the other enabled vehicles are and establish one or many MO Partners. This is because MO2 and both MO3 methods require each enabled vehicle to know when they pass one another. Remember, enabled vehicles have the following capabilities:

- a DSRC compliant radio,
- the ability to count the number of times the vehicle passes another vehicle or is passed by another vehicle,
- the ability to track its own location and other vehicles' locations on a digital map.

Upon nearing a test area, an enabled vehicle will begin to regularly broadcast an MPR using UDP over IPV6. UDP is not reliable; however, for this application, it does not need to be. The vehicle is simply advertising itself as an enabled vehicle. Any other enabled vehicles will then respond with an MO Partner Request. This will be discussed in the next section.

TABLE 8: The MPR Format

Field	required	size
Test Area ID	required	2 bytes
MO Method Type	required	3 bits
Temporary ID	required	4 bytes

The Payload of the MPR message has the following format that is also shown in Table 8:

Test Area ID - The Test Area ID is the ID value from the TAA message. It is 2 bytes and represents the Test Area being to be measured.

MO Method Type - The MO Method Type field identifies which MO method that the vehicle should utilize. Additional details can be see in the TAA message section above. The MO Method Type is represented by 3 bits. The value mapping can be see in the TAA message section above.

Temporary ID - Temporary ID is the ID the vehicle will use throughout the test area when sending BSM messages. This ensures any future MO Partners will be able to track its location from these BSM messages. The Temporary ID field is represented by 4 bytes.

This completes the MPR message format.

A possible alternative is for the enabled vehicles to register with the RSU and the RSU includes a list of enabled vehicles in the WSM. This method requires complete RSU coverage.

3.3.4 MO PARTNER ACCEPT

The second of the two MO Partner Negotiation messages is the MO Partner Accept (MPA) message. When an enabled vehicle receives an MPR message it will respond with a MPA message to establish themselves as MO Partners. MO Partners have responsibilities that will be discussed in more details in the appropriate MO chapters that follow.

The MPA message is sent using TCP over IPV6 because it is important that both vehicles are aware they are MO partners. TCP offers the required reliability.

TABLE 9: MPA Format

Field	required	size
Test Area ID	required	2 bytes
Start Time	required	4 bytes
MO Method Type	required	4 bits
Temporary ID	required	4 bytes

The Payload of the MPA message has the following format that is also shown in Table 9:

Test Area ID - The Test Area ID is the ID value from the TAA message. It is 2 bytes and

represents the Test Area being to be measured.

Start Time - The Start Time is the time the method started. Typically this is the same time as when the message was sent. The Time fields use the DTime field from the DSRC specification [48] which contains the three fields: DHour, DMinute, and DSecond. DHour is a 1 byte field containing the hour from 0 to 23. DMinute is a 1 byte field containing the minute from 0 to 59. DSecond is a 2 byte field containing the microseonds from 0 to 60,000.

MO Method Type - The MO Type is field that identifies the MO Type the accepting vehicle will use. This is the same MO Type from the TAA message, and it should match the MO Type sent in the MPR message. See the TAA message for more details, including the mapping of values to Methods. The MO Method Type field has a size of 3 bits.

Temporary ID - Temporary ID is the ID the vehicle will use throughout the test area when sending BSM messages. This ensures any future MO Partners will be able to track its location from these BSM messages. The Temporary ID has a size of 4 bytes.

This completes the MPA message format.

3.3.5 TALLY EXCHANGE MESSAGE

At certain times within the MO Methods, the pass pair must exchange tallies. This is done with the Tally Exchange Message (TEM). These messages are sent using TCP over IPV6, again TCP is used for this message because it is a reliable protocol.

Each of the MO1, MO2, and MO3 methods utilize the Tally Exchange Protocol.

The Payload of the TEM message has the following format that is also shown in Table 10:

Test Area ID - The Test Area ID is a unique identifier that represents the test area. This is the same ID that was originally sent in the TAA message.

Start Time - The Start Time field represents the start time of the tally window. The time is broke into three subfield, DHour, DMinute, and DSecond. The DHour field contains an integer value for the hour between 0 and 23. The DMinute field contains an integer value for the minute between 0 and 59. The DSecond field contains an integer value for the micro second between 0 and 59,999.

End Time - The End Time field represents the end time of the tally windows. The time in also broken into the same three subfields as startTime.

Start Location - The Start Location field represents the starting location of the tally window. It uses the Position2D field type which includes the latitude and longitude values. The precision of each field is 1/8th micro degrees. Both fields together use 8 bytes.

TABLE 10: Fields in the proposed Tally Exchange Message

Field	Field Type	size
Test Area ID	TestAreaID	2 bytes
Start Time	DTime	4 bytes
End Time	DTime	4 bytes
Start Location	Position2D	8 bytes
End Location	Position2D	8 bytes
Heading	Heading	2 bytes
Tally Type	Bit	1 bit
Tally Codirectional	Tally	1.5 bytes
Tally Oncoming	Tally	1.5 bytes
Flow	Flow	2 bytes
Density	Density	2 bytes

End Location - The End Location field represents the ending location of the tally window. It also uses the same Position2D field type as the starting location.

Heading - The Heading field is the direction of travel for the vehicle. Next is the Heading field which represents the direction the vehicle is currently traveling. The field is 2 bytes, which each value representing a unit of 360/32,768 degrees

Tally Type - The Tally Type field is a field used to specify the phase of the MO method. This is only used in the MO3-Flow method. It will be 0 for the first phase and 1 for the second phase. For other methods, this field will be set to a 0.

Tally Codirectional - The Tally Codirectional field is the codirectional tally, that is, the tally in the same direction as the vehicle. It uses a new Tally field type that contains a 1.5 bytes (12 bits) integer between 0 and 4,095.

Tally Oncoming - The Tally Oncoming field is the oncoming tally, that is, the tally in the opposite direction as the vehicle. It also uses the new Tally field type that contains a 1.5 bytes (12 bits) integer between 0 and 4,095.

Flow - The Flow field is the estimated flow as determined by the MO method. The Flow represents vehicles per hour and is multiplied by 10 before encoding. Being 2 bytes long, it

can represent a flow between 0.0 and 6,653.5 with 1 digit of accuracy.

Density - The Density field is the estimated density as determined by the MO method. The density represents the density in km; however, is multiplied by 100 before encoding. Being 2 bytes long it can represents a density between 0.00 to 665.35 with 2 digits of accuracy.

This completes the TEM message format.

3.3.6 USE OF BSM IN MO

In the MO methods, it is critical to know the moment two enabled vehicles pass one another. The MO Partner Negotiation established two vehicles as MO Partners. Through the MPR and MPA messages, each will know the Temporary ID of the other. Each will then track the other using the details from the BSM. The BSM provides a temporary Id, position, heading, and velocity at a rate of ten times a second. Using the method in Section 3.2.1, this information can be used to determine when two enabled vehicles pass one another.

It is important to note that while in the test area, the Temporary ID of the vehicle should not change. If it does change, its MO partners will not be able to track its locations through BSM messages.

CHAPTER 4

MOBILE OBSERVER METHOD

This chapter describes the Mobile Observer (MO) method, which is a modernized variant of the Moving Observer method described by Wardrop and Charlesworth [26]. The name Mobile Observer is chosen to capture the modernization of the method. When referring to the MO method, this will reference the Mobile Observer method we propose. When referring to the original method, we will also refer to it as the Moving Observer method.

Before describing the first of the MO method, we first introduce each of the methods and lay a foundation of the methods with some prelimary terminology and results.

4.1 INTRODUCTION TO THE MOBILE OBSERVER METHODS

The goal of each method described in this and the following chapters is to measure the flow or density of a particular stretch of road. This stretch of road will be called the *test area*. For each method, it is important that the number of vehicles is conserved between the entrance of the test area and the exit of the test area. Two possible candidates for such a test area are a stretch of road between two on-ramps or off-ramps on a highway, or the stretch of road between two intersections on an urban street, again assuming there are no exits. Depending of the method, there may or may not be bi-directional traffic. Each of the methods rely on designated vehicles, using their on-board sensing capabilities, to maintain a *tally* of the difference between the number of times other vehicles pass them and the number of times they pass other vehicles. A vehicle is said to be *enabled* if it has:

- a V2X compliant radio,
- the ability to count the number of times the vehicle passes another vehicle or is passed by another vehicle,
- the ability to track its own location on a digital map.

There are also requirements for the roadway infrastructure to support the methods. First, the infrastructure must have the ability to notify an enabled vehicle of the location of the test area. Second, there must be an RSU or local agent with the ability to aggregate results and disseminate the results to other vehicles.

The three methods covered in this and the following chapters use the concept of this tally in novel ways. The three methods are:

1. MO 1 - a modern update to the Wardrop and Charlesworth [26] moving observer method that utilizes the capabilities of modern day vehicles to negate some of the shortcomings of the original method.

2. MO 2 - a method that uses the same tallies as MO1. It is used to determine density based on two codirectional vehicles that pass within the test area. This method is suitable for highways and long roads such as the stretch of road between two exits.

3. MO 3 - a method similar to MO2, but one of the test pairs is in the oncoming traffic. This method is suitable for urban roads such as the road between two intersections. The MO 3 method is subdivided into another method that measures flow instead of density.

In order to describe our method, we need to establish terminology and to prove a few technical results that will be used as stepping stones towards establishing our main result.

4.2 PRELIMINARIES OF OUR METHODS

This work assumed a multi-lane roadway populated by vehicles that travel in both directions. For consistency the terms codirectional vehicles or codirectional traffic will be used for vehicles or traffic that travels in the same direction. The terms oncoming vehicles or oncoming traffic will be used for vehicles that travel in opposite directions. The vehicles do not need to travel at constant velocity. In fact, we allow them to vary their velocity in arbitrary ways and for all sorts of reasons. For technical reasons, we assume that vehicle velocities are expressed in the form of *infinite-precision* real numbers. An important corollary of this assumption is that no two vehicles travel at exactly the same velocity.

In order to describe our protocol, we need to establish terminology and to prove a few technical results. These technical results will be used as stepping stones to establish the main results of each of the methods.

We will denote vehicles by capital letters, $X,\ Y,\ Z$, etc. We think of vehicles, regardless of the lane they occupy and of the direction in which they move, as points on the positive real axis. Naturally, these points may move left to right or right to left. At time $t,\ (t \geq 0)$, we associate with a generic vehicle X its coordinate $x(t)$ on the real axis.

Definition 1. *Given* codirectional *vehicles X and Y, vehicle Y is said to be* behind *vehicle X at time t if exactly one of the following conditions is satisfied:*

- $y(t) < x(t)$;
- $y(t) = x(t)$ *and there exists a positive real* δ *such that for all* ϵ, $(0 < \epsilon < \delta)$, $y(t-\epsilon) < x(t-\epsilon)$.

Refer to Figure 9 for illustration. At time $t - \epsilon$ and at time t, Y is behind X.

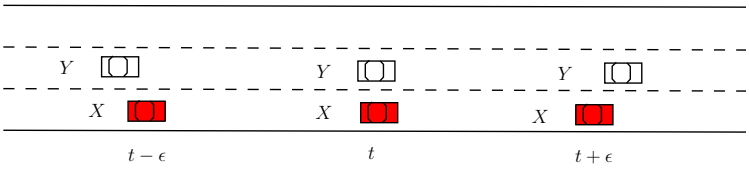


Fig. 9: Vehicle Y passes vehicle X at time t.

If the first condition of Definition 1 holds, we say that Y is *strictly* behind X at time t. The second condition of Definition 1 states that Y is behind X at time t if, for some suitably chosen $\delta > 0$, Y was strictly behind X during the interval $(t - \delta, t)$ and "drew even" with X at time t.

Next, observe that Definition 1, together with the fact that no two vehicles travel at the same velocity, guarantees that given t, $(t > 0)$, and two arbitrary vehicles on the roadway, exactly one of them is behind the other. The formal statement of this intuitive property is captured by Lemma 4.2.1.

Lemma 4.2.1. *If no two vehicles travel at the same velocity then exactly one is behind the other.*

Proof. Consider two vehicles X and Y moving left to right on the real axis. Let t, $(t > 0)$, be arbitrary. We need to show that at time t, exactly one of X and Y is behind the other. At time t, the coordinates of these vehicles are $x(t)$ and $y(t)$. If $y(t) < x(t)$ then, by Definition 1, Y is behind X at time t.

Assume, therefore, that $y(t) = x(t)$ and let $t' \leq t$ be the *last* time that $y(t') = x(t')$ occurred. We write

$$\delta = \begin{cases} t & \text{if} \quad t' = t; \\ t - t' & \text{if} \quad t' < t. \end{cases} \tag{16}$$

Our choice of t' and δ guarantees that for every $s,\ s \in (t - \delta, t)$, either $x(s) < y(s)$ or $y(s) < x(s)$. Thus, by the second part of Definition 1, exactly one of X or Y is behind the other at time t, as claimed. □

It is intuitively clear that the behindness property is *transitive*. A formal statement of this intuitive property follows.

Lemma 4.2.2. *The behindness property is transitive. If at time t, X is behind Y and Y is behind Z, then X is behind Z.*

Proof. Let $x(t),\ y(t),\ z(t)$ be the coordinates, at time t, of X, Y, and Z respectively. If either X is strictly behind Y or if Y is strictly behind Z then, clearly, X is strictly behind Z.

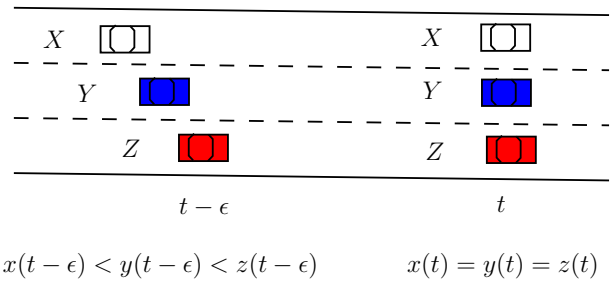


Fig. 10: Illustrating transitivity of the behindness property.

Assume, therefore, that $x(t) = y(t) = z(t)$ and refer to Figure 10. By Definition 1, since X is behind Y, there must exist a positive $\delta_{XY} > 0$ such that for all $\epsilon, (0 < \epsilon < \delta_{XY})$, $x(t - \epsilon) < y(t - \epsilon)$. Similarly, since Y is behind Z, at time t, there must exist a positive $\delta_{YZ} > 0$ such that for all $\epsilon, (0 < \epsilon < \delta_{YZ})$, $y(t - \epsilon) < z(t - \epsilon)$.

Let $\delta = \min\{\delta_{XY}, \delta_{YZ}\}$. With this choice of δ, for every $\epsilon, (0 < \epsilon < \delta)$, we have $x(t - \epsilon) < y(t - \epsilon) < z(t - \epsilon)$, confirming that X is behind Z, as claimed. □

Definition 1 can be extended as follows: given a time interval J, we say that Y is behind X in J if for every $t, (t \in J)$, Y was behind X at time t.

Definition 2. *Refer again to Figure 9; given* codirectional *vehicles X and Y, we say that vehicle Y* passes *vehicle X at time t, $(t > 0)$, if there exists a positive δ such that for every $\epsilon, (0 < \epsilon < \delta)$, Y was behind X in the interval $[t - \epsilon, t]$ and X was behind Y in $(t, t + \epsilon]$.*

It is important to note that Definitions 1 and 2, combined, imply that if vehicle Y passes X at time t, Y is still behind X at the time of passing, while X falls behind Y immediately thereafter.

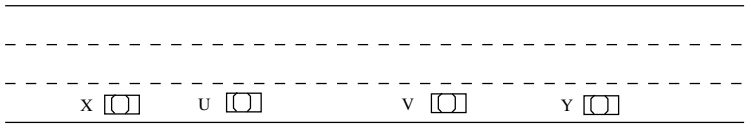


Fig. 11: Illustrating the transitivity of the betweenness relation.

Consider an interval $I = [t_1, t_2]$ with $t_1 < t_2$. Definition 2 guarantees that Y can pass X at every *interior* point of I. It is also possible for Y to pass X at t_1: this is so because Y is still behind X at t_1 and the transition from Y being behind X to X being behind Y takes place within I. However, even if Y passes X at t_2, this passing does not occur in I since the transition from Y being behind X to X being behind Y does not occur within I.

Definition 3. *Given* codirectional *vehicles X, Y, and Z, we say that vehicle Z is* between *Y and X at time t if Y is behind Z, and Z is behind X at time t.*

Note, in particular, that if Y passes Z, and Z passes X at time t then, according to Definition 3, Z is between Y and X (in that order) at time t. Refer to Figure 12 for an illustration of this important special case. We note that even though, in practice, the probability of three vehicles passing each other at the same exact time is very small, it is not *zero* and therefore must be considered.

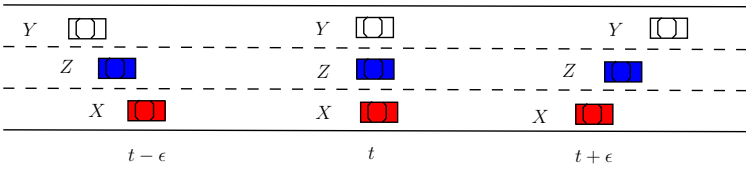


Fig. 12: Y passes Z and Z passes X at time t.

It is easy to confirm that the betweenness property is transitive in the following natural way: as illustrated in Figure 11, if U is between X and V and V is between U and Y, then V is between U and Y and, also, U and V are between X and Y. It will be shown that betweenness is important to approximating traffic density and flow.

With this in mind, given two vehicles X and Y and assuming X behind Y at time t, we let $B(X, Y, t)$ denote the set of vehicles that happen to be *between* X and Y at time t. Importantly, X and Y are not in the set $B(X, Y, t)$. The cardinality of $B(X, Y, t)$, that is, the number of vehicles between X and Y at time t will be denoted by $b(X, Y, t)$. Refer to Figure 13 depicting a roadway segment at time t. Here, X is passing T, T is behind U, U is behind V, V is behind W and, W is behind Z. Finally, Z is about to pass Y. Now, the transitivity of the betweenness relation implies that T, U, V, W, Z are *between* X and Y at time t and so $B(X, Y, t) = \{T, U, V, W, Z\}$, while $b(X, Y, t) = |B(X, Y, t)| = 5$.

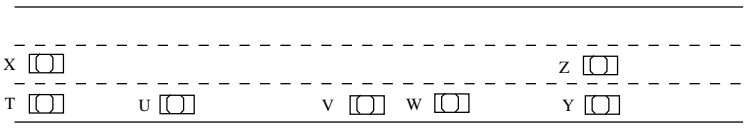


Fig. 13: Illustrating the set $B(X, Y, t)$ of vehicles between X and Y.

Consider a time interval $I = [t_1, t_2]$ with $t_1 < t_2$. Given a distinguished vehicle X, we classify the remaining vehicles participating in the traffic as follows.

Definition 4. *Vehicle Y is of Type 1 with respect to X in I if Y was behind X at t_1 and X was behind Y at t_2.*

Refer to Figure 14 for an illustration. Observe that Definition 4 requires Y to pass X at least once in I; as observed above, Y may pass X at any time in I, except at t_2.

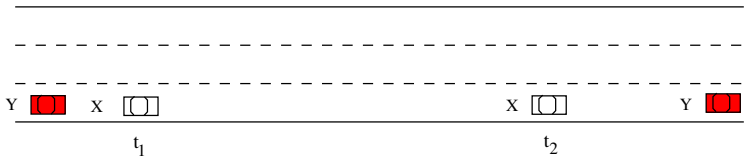


Fig. 14: Y is of Type 1 with respect to X in I.

Definition 5. *Vehicle Y is of Type 2 with respect to X in I if Y was behind X at both t_1 and t_2.*

Refer to Figure 15 for an illustration. Observe that Y may or may not pass X in I.

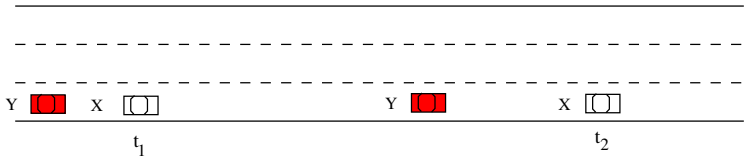


Fig. 15: Y is of Type 2 with respect to X in I.

Definition 6. *Vehicle Y is of Type 3 with respect to X in I if X was behind Y at both t_1 and t_2.*

Refer to Figure 16 for an illustration. Observe that X may or may not pass Y in I.

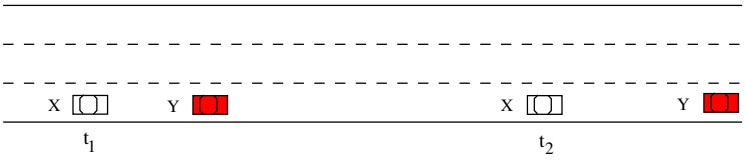


Fig. 16: Y is of Type 3 with respect to X in I.

Definition 7. *Vehicle Y is of Type 4 with respect to X in I if X was behind Y at t_1 and Y was behind X at t_2.*

Refer to Figure 17 for an illustration. Observe that Definition 7 requires X to pass Y at least once in I; as observed above, X may pass Y at any interior point of I, but not at t_2.

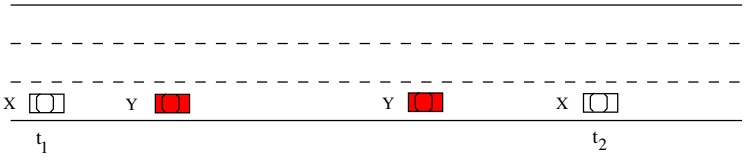


Fig. 17: Y is of Type 4 with respect to X in I.

Note: Observe that Definitions 4, 5, 6, and 7 have a number of interesting consequences. In particular, there is a certain duality between vehicles of Types 2 and 3 as well as between those of Types 1 and 4. Specifically,

- Y is of Type 2 with respect to X in I if and only if X is of Type 3 with respect to Y in I. Observe also that if X is of Type 2 or 3 with respect to Y , then X need not pass Y in I. If it does, however, then Y must also pass X;

- Similarly, Y is of Type 3 with respect to X in I if and only if X is of Type 2 with respect to Y in I. Observe also that if Y is of Type 2 or 3 with respect to X, then Y need not pass X in I. If it does, however, then X must also pass Y;

- Y is of Type 1 with respect to X in I if and only if X is of Type 4 with respect to Y in I.
- Similarly, Y is of Type 4 with respect to X in I if and only if X is of Type 1 with respect to Y in I.

Definition 8. *Consider a generic vehicle X and let $n_f(X, I)$ denote the number of times X was passed by other codirectional vehicles in I. Similarly, let $n_s(X, I)$ stand for the number of times X passed other codirectional vehicles in I. We write*

$$\tau(X, I) = n_f(X, I) - n_s(X, I) \tag{17}$$

and refer to $\tau(X, I)$ as the tally *of X over the time interval I.*

Lemma 4.2.3. *Every vehicle of Type 1 with respect to X in I contributes $+1$ to $\tau(X, I)$.*

Proof. Consider an arbitrary vehicle Y of Type 1 with respect to X in I. Every time Y passes X in I, we record a $+1$ and every time X passes Y in I we record a -1. Since Y is of Type 1, the sequence of passings is an alternating sequence of $+1$s and -1s starting with a $+1$ and ending with a $+1$. It follows that this alternating sequence contains one more $+1$s than -1s. Thus, Y contributes *exactly* $+1$ to $\tau(X, I)$ and the proof is complete. □

Lemma 4.2.4. *Every vehicle of Type 2 or Type 3 with respect to X in I contributes* 0 *to $\tau(X, I)$.*

Proof. We will prove the statement for a vehicle of Type 2. The proof for a Type 3 vehicle is similar.

Let Y be of Type 2 with respect to X in I. As in the proof of Lemma 4.2.3, we model the passing between Y and X as a sequence of $+1$s and -1s. First, if Y does not pass X in I, then it contributes 0 to $\tau(X, I)$. Assume, therefore, that Y passes X at least once in I. Since, by definition, at t_2, Y is behind X, in their last encounter X must have passed Y. Thus, the corresponding alternating sequence begins with a $+1$ and ending with a -1 and so the number of $+1$s and -1s must agree. Therefore, the contribution of Y to $\tau(X, I)$ must be 0, as claimed. □

Lemma 4.2.5. *Every vehicle of Type 4 with respect to X in I contributes -1 to $\tau(X, I)$.*

Proof. Consider an arbitrary vehicle Y of Type 4 with respect to X in I. Since, by definition, Y is behind X at t_2, the first encounter between them must be a -1. Similarly, their last passing must be a -1. It follows that the sequence of passings can be modeled as an alternating sequence of $+1$s and -1s starting and ending with a -1. Thus, Y contributes *exactly* -1 to $\tau(X, I)$, as claimed. □

These definitions and lemmas provide the basic building blocks that will be used for the methods introduced throughout this chapter.

4.3 MOBILE OBSERVER 1

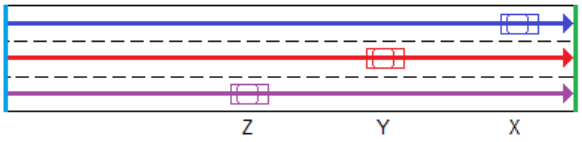
The Mobile Observer 1 method is a modern update to the method proposed by Wardrop and Charlesworth [26] Their method, known as the *Moving Observer* method, involves a test vehicle traversing a given test area both in the direction of the traffic whose parameters are of interest and also in the opposite direction.

The *Moving Observer* method has many drawbacks that our MO method negates. First, in the *Moving Observer* method, as pointed out by several authors [27, 28, 29], the test vehicle must make several passes of the test area to get an accurate reading, in which time the traffic may change. Additionally, since this method relies on sensors to detect oncoming traffic, in cases of occlusion between the oncoming traffic, the method may not work.

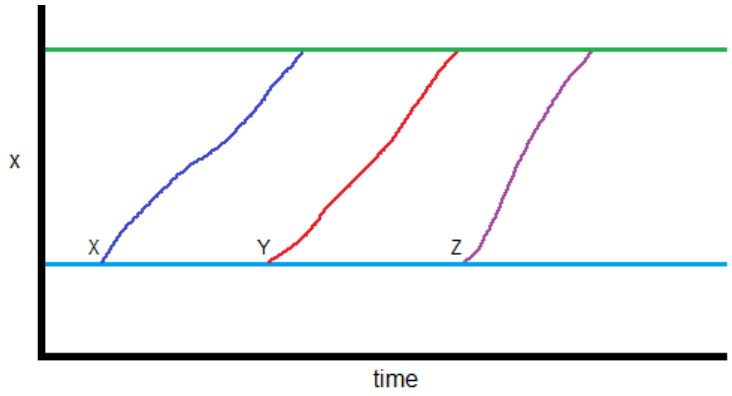
In our modern update of the *Moving Observer* method we take advantage of the novel on-board sensing, computing, and networking capabilities in present-day vehicles, thus eliminating the need for a test vehicle to perform several runs in both directions; in fact, our MO method only involves vehicles codirectional to direction of interest.

Our variant of the MO method only requires vehicles to detect the number of vehicles they pass and the number of vehicles that pass them on a given road segment. This can be easily done using either short-range on-board radar devices or, perhaps, specialized camera-based sensors [31]; we refer the reader back to Section 3.2. The availability of these on-board devices, in effect, make each vehicle in the traffic behave as a test vehicle acting independently of other vehicles. By aggregating the collected information either locally or centrally, the fundamental traffic parameters can be determined.

Use Figure 18 as an illustration of the MO1 method. There are two figures representing two different views of the traffic. In Figure 18(a) shows a top down view of vehicles X, Y, and Z on the road and Figure 18(b) shows the time-space view of the vehicles. In the figure, there are three vehicles, X, Y, and Z, with non-constant velocities all traveling in the same



(a) Roadway view



(b) Time-space view

Fig. 18: Illustrating the MO1 method.

direction. As each vehicle enters the test area, represented by the light blue line, it begins recording its tallies. Upon exiting the test area, represented by the green line, it will send its tallies to other vehicles or a road side unit to be aggregated. In the figures, X will enter the roadway first and maintain a tally until it exits the test area. Upon exiting it will send its tally to another vehicle Y that has just entered the test area. In the test area, Y will maintain its own tally and when it exits the test area it will do two things. First it will use the tallies from X and itself to estimate the traffic parameters of the test area. Second it will send its own tally to vehicle Z that is just about to enter the test area. Vehicle Z will then enter the test area and maintain its own tally until it exits the test area. It will use the tallies from vehicle Y and itself to estimate the traffic parameters of the test area.

4.3.1 TECHNICALITIES

Consider a test area that is a road segment AB of length l. A vehicle X that enters the test area at time t_1 and exits at time t_2. The interval $I = [t_1, t_2]$. Let $n_f(X, I)$ denote

the number of times vehicle X is passed by other vehicles in interval I. Similarly, let $n_s(X, I)$ stand for the number of times vehicle X passes other vehicles in interval I. Further, let $\overline{n}_i(X, I)$ and $\underline{n}_i(X, I)$ denote, respectively, the number of times a vehicle of Type i, $(1 \leq i \leq 4)$, has passed and was passed by vehicle X in interval I.

In this notation, it is clear that

$$n_f(X, I) = \sum_{i=1}^{4} \overline{n}_i(X, I) \tag{18}$$

and, likewise,

$$n_s(X, I) = \sum_{i=1}^{4} \underline{n}_i(X, I). \tag{19}$$

Lemma 4.2.3 has the following important consequence.

Corollary 4.3.1. *The number, $n_1(X, I)$, of Type 1 vehicles satisfies*

$$n_1(X, I) = \overline{n}_1(X, I) - \underline{n}_1(X, I).$$

Proof. By Lemma 4.2.3, each vehicle of Type 1 contributes exactly 1 to $\tau(X, I) = n_f(X, I) - n_s(X, I)$. On the one hand, the total contribution of all vehicles of Type 1 is n_1; on the other hand, this contribution must be $\overline{n}_1(X, I) - \underline{n}_1(X, I)$. The conclusion follows. □

Lemma 4.2.4 has the following important consequence.

Corollary 4.3.2. *$\overline{n}_2(X, I)$, $\underline{n}_2(X, I)$, $\overline{n}_3(X, I)$ and $\underline{n}_3(X, I)$ satisfy, respectively, $\overline{n}_2(X, I) - \underline{n}_2(X, I) = 0$ and $\overline{n}_3(X, I) - \underline{n}_3(X, I) = 0$.*

Proof. By Lemma 4.2.4, each vehicle of Type 2 (resp. Type 3) contributes 0 to $\tau(X, I) = n_f(X, I) - n_s(X, I)$. Thus, the total contribution of all vehicles of Type 2 (resp. Type 3) is $\overline{n}_2(X, I) - \underline{n}_2(X, I)$ (resp. $\overline{n}_3(X, I) - \underline{n}_3(X, I)$) and must be 0. □

Lemma 4.2.5 has the following important consequence.

Corollary 4.3.3. *The number, $n_4(X, I)$, of Type 4 vehicles satisfies*

$$n_4(X, I) = \underline{n}_4(X, I) - \overline{n}_4(X, I).$$

Proof. By Lemma 4.2.5, each vehicle of Type 4 contributes exactly -1 to $\tau(X, I) = n_f(X, I) - n_s(X, I)$. On the one hand, the total contribution of all vehicles of Type 1 is $n_4(X, I)$; on the other hand, this contribution must be $(-1) \times (\overline{n}_4(X, I) - \underline{n}_4(X, I)) = \underline{n}_4(X, I) - \overline{n}_4(X, I)$, as claimed □

Lemma 4.3.4. *The following identity holds*

$$\tau(X, I) = n_1(X, I) - n_4(X, I). \tag{20}$$

Proof. By Equations (18) and (19) we write

$$\begin{aligned}
\tau(X, I) &= n_f(X, I) - n_s(X, I) \\
&= \sum_{i=1}^{4} \overline{n}_i(X, I) - \sum_{i=1}^{4} \underline{n}_i(X, I) \\
&= \sum_{i=1}^{4} (\overline{n}_i(X, I) - \underline{n}_i(X, I)) \\
&= (\overline{n}_1(X, I) - \underline{n}_1(X, I)) + (\overline{n}_2(X, I) - \underline{n}_2(X, I)) \\
&\quad + (\overline{n}_3(X, I) - \underline{n}_3(X, I)) + (\overline{n}_4(X, I) - \underline{n}_4(X, I)) \\
&= n_1(X, I) + 0 + 0 - n_4(X, I) \\
&\quad \text{[by Corollaries 4.3.1, 4.3.2, 4.3.3]} \\
&= n_1(X, I) - n_4(X, I),
\end{aligned}$$

□

Let q and κ be, respectively, the traffic flow and traffic density over the road segment AB. For this method, we assume that q and κ are constant; therefore, we simplify the notation and will not define the values of q and κ by the interval. Additionally, values of τ, n_f, n_s, and n_i will be in reference to a single test vehicle and its notation will also be simplified without referring to the specific vehicle or interval. For example, $\tau(X, I)$ will be referred to as τ in the context of a test vehicle over the road segment AB. The next result, Theorem 4.3.5, expresses τ in terms of q and κ.

Theorem 4.3.5. *Let t be the time it takes the test vehicle to traverse the road segment AB of length l. Then*

$$\tau = tq - \kappa l. \tag{21}$$

Proof. Consider, again the road segment AB of length l that our test vehicle enters at time t_1. We are interested in evaluating the number of vehicles in AB at time t_2, namely when the test vehicle has reached B. Note that $t = t_2 - t_1$. On the one hand, this number must be κl. On the other hand, the same number of vehicles consists of:

- all the vehicles of Type 2, that is, all the vehicles that entered AB in the time interval (t_1, t_2) but have not exited by time t_2. This number is $tq - n_1$, where n_1 is the number of Type 1 vehicles;

- all the vehicles of Type 4, that is, all the vehicles that were in AB at time t_1 and have not left by time t_2. In our notation, this number is n_4.

It follows that

$$\begin{aligned}\kappa l &= (tq - n_1) + n_4 \\ &= tq - (n_1 - n_4) \\ &= tq - (n_f - n_s), \quad \text{[by Lemma 4.3.4]} \\ &= tq - \tau,\end{aligned}$$

from where the statement of the theorem follows directly. □

4.3.2 DISCUSSION AND EXTENSIONS

The main goal of this subsection is to take a closer look at Equation (21) and its corollaries.

We begin by noting that, to the best of our knowledge, Equation (21) was derived in the literature under the assumption of the traffic consisting of a number of sub-flows involving vehicles moving at constant velocitu. Our derivation removes this restriction and is general. In particular, this yields a general proof of Wardrop and Charlesworth's original Moving Observer method.

Let $v = \frac{l}{t}$ be the average velocity of our test vehicle over AB. Let v_{avg} be the average vehicular velocity over AB. By dividing both sides of Equation (21) by t one obtains

$$\begin{aligned}\frac{n_f - n_s}{t} &= q - \kappa\, v \qquad (22)\\ &= \kappa\, v_{avg} - \kappa\, v \\ &= \kappa\,(v_{avg} - v)\,. \qquad (23)\end{aligned}$$

Notice that, by Equation (23), $n_f - n_s = 0$ if and only if $v = v_{avg}$. If $n_f - n_s = 0$ the test vehicle is said to be "floating" since the number of times it is passed by a vehicle in AB matches the number of times it passes a vehicle in AB. As it turns out, a popular variant of the MO method proposed in the literature [27], involves driving the test vehicle so as to ensure that $n_f - n_s = 0$, which, as we saw, guarantees that the test vehicle's velocity is an approximation of the average vehicular velocity over AB [29, 28].

Interestingly, the left-hand and right-hand sides of Equation (22) are measured in vehicles/hour and, therefore, represent flows. The flow $\frac{n_f - n_s}{t}$ is positive if the number of times

the test vehicle is passed exceeds the number of times it passes a vehicle. This, of course, corresponds to the case $v < v_{avg}$. Similarly, the flow $\frac{n_f - n_s}{t}$ is *negative* in case the test vehicle moves faster than the average vehicular velocity.

4.3.3 THE MO1 ALGORITHM

The goal of this section is to propose a variant of the MO method that allows the determination of the fundamental traffic parameters, flow, density and average velocity without the need for the test vehicle to perform several runs in both directions as Wardrop and Charlesworth's original MO method does.

To motivate our variant of the MO method, we note that as long as the test vehicle can count the number of times it was passed and the number of times it passed other vehicles in AB, it can compute $n_f - n_s$. In addition, it knows t and l. Therefore, Equation (21) contains two unknowns, namely q and κ, while $n_f - n_s$, t and l are known. In order to determine the two unknowns, a second equation involving q and κ is needed. Such an equation can be obtained, as in Wardrop and Charlesworth's original method [26], by having the test vehicle traverse AB in the direction from B to A, counting the number of vehicles encountered as well as the time of traversal. However, we will not pursue this, because we have found an approach more suitable for this day and age.

Instead, we propose to obtain a second equation involving q and κ by exchanging information among the vehicles traversing the same segment AB. For this purpose, we assume that the vehicles in the traffic are equipped with on-board GPS, a digital map, short-range on-board radar devices, and a radio transceiver. While today on-board radar devices are only found in high-end vehicles, we expect them to become commonplace in a few years. We note that using its on-board radar devices, each vehicle can detect the number of times it passes other vehicles as well as the number of times it is passed by other vehicles in a given road segment. With this assumption, every vehicle on the road can act as a test vehicle.

Acting independently, all vehicles compute their own local version of $n_f - n_s$, t and l and set up their own version of Equation (21) where q and κ are shared unknowns, since all the vehicles witness the same traffic conditions.

It is clear that by exchanging its own Equation (21) with that of a suitably chosen neighboring vehicle, each vehicle can obtain a system of equations in q and κ. By solving this system, each vehicle can determine flow q and density κ.

Indeed, consider a pair of vehicles that travel at *different* velocity, and let t and t' be, respectively, the time they take to traverse road segments of lengths l and l'. Let $n_f - n_s$

and $n'_f - n'_s$ be the corresponding "tallies" computed by the two vehicles. By exchanging information, each of these vehicles will set up the following system of equations

$$\begin{cases} n_f - n_s = tq - l\kappa \\ \\ n'_f - n'_s = t'q - l'\kappa. \end{cases} \tag{24}$$

It is easy to confirm that this system admits of a unique solution if and only if

$$tl' \neq t'l \tag{25}$$

which is equivalent to saying that the two vehicles not travel at the same average velocity. Under this mild condition, the system (24) can be solved to yield

$$q = \frac{l'(n_f - n_s) - l(n'_f - n'_s)}{l't - lt'}. \tag{26}$$

and

$$\kappa = \frac{t'(n_f - n_s) - t(n'_f - n'_s)}{l't - lt'}. \tag{27}$$

We claim that

Lemma 4.3.6. *If Equation (25) holds, then*

$$\frac{n_f - n_s}{t} \neq \frac{n'_f - n'_s}{t'} \tag{28}$$

and

$$\frac{n_f - n_s}{l} \neq \frac{n'_f - n'_s}{l'}. \tag{29}$$

Proof. To show that Equation (28) must hold, let $v = \frac{l}{t}$ and $v' = \frac{l'}{t'}$ be the average velocities of two vehicles. From Equation (22) it follows that

$$\frac{n_f - n_s}{l} - \frac{n'_f - n'_s}{l'} = k(v - v').$$

Since, as we saw, Equation (25) is equivalent to $v \neq v'$, (28) follows.

Next, we propose to establish the contrapositive of Equation (29). For this purpose, suppose that

$$\frac{n_f - n_s}{l} = \frac{n'_f - n'_s}{l'}.$$

Since $l = v\ t$ and $l' = v'\ t'$, the above equality becomes

$$\frac{n_f - n_s}{v\ t} = \frac{n'_f - n'_s}{v'\ t'}.$$

By using Equation (22), again, we obtain

$$\frac{q-\kappa v}{v} = \frac{q-\kappa v'}{v'}$$

which, in turn implies $v = v'$ or, equivalently, Equation (25). Thus, Equation (29) is satisfied as well. This completes the proof of the lemma. □

Lemma 4.3.6 confirms that if the system of Equations (24) is derived from data collected by two vehicles traveling at *different* velocity, then the values of q and κ are non-zero. In turn, this allows the computation of the average vehicular velocity, v_{avg}, by using Equation (3).

Indeed, by employing Equations (25), (26) and (27), we write

$$\begin{aligned} v_{avg} &= \frac{q}{\kappa} \\ &= \frac{\frac{l'(n_f-n_s)-l(n'_f-n'_s)}{l't-lt'}}{\frac{t'(n_f-n_s)-t(n'_f-n'_s)}{l't-lt'}} \\ &= \frac{l'(n_f-n_s)-l(n'_f-n'_s)}{t'(n_f-n_s)-t(n'_f-n'_s)}. \end{aligned} \tag{30}$$

The accuracy of our variant of the MO method can be further enhanced as follows. Instead of each vehicle collecting traffic data from one single vehicle witnessing the same traffic conditions, it can collect data from a number of such vehicles. By averaging the results, average values $\hat{q}$, $\hat{\kappa}$ and $\hat{v}_{avg}$ can be computed and used as estimates for the fundamental traffic parameters.

One last issue that needs to be addressed is the performance of our variant of the MO method at low penetration rate, namely, under conditions where the number of vehicle that have on-board capabilities that allow them to compute $n_f - n_s$ is relatively low. At low penetration rate, the suitably enabled vehicles cooperate, as described before, to estimate the fundamental traffic parameters and the resulting values are then disseminated to the other vehicles using their on-board radios.

4.4 MO1 SIMULATION AND RESULTS

4.4.1 SIMULATION MODEL

We tested our MO method by applying it to vehicles in a freeway type model. We assume a long, straight section of road where no vehicles may enter or exit, except at the

beginning or end of the road. We also assume there are enough lanes such that vehicles do not adjust their velocity to pass one another.

Two points are marked along the road, point A and point B, with a distance of 5km apart. At point A vehicles start counts for n_f and n_s. We assume the vehicle has the capability to determine these counts.

At point B, the vehicle communicates τ, l, $n_f - n_s$ with each of its neighbors. Vehicles in front of the test vehicle have passed point B and have already communicated τ, l, $n_f - n_s$ with the test vehicle. The test vehicle saved these values in memory until this point. The vehicle will solve Equations (26), (27) and (30) using up to 10 vehicles in front of it, sorted by most recent vehicle first, to get values for q, κ and v_{avg}. The median values of each are saved and transmitted to a database.

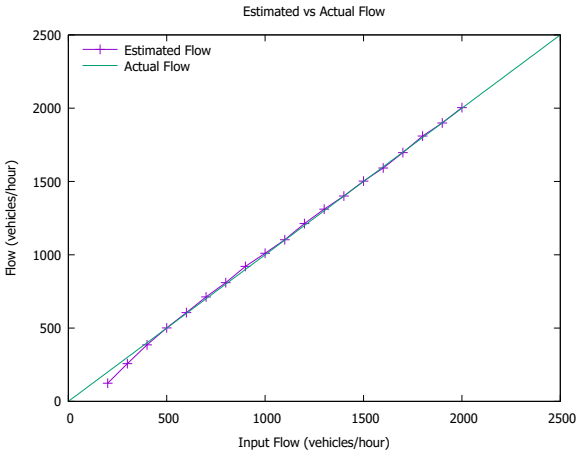
The simulation runs for a total 15 minutes, with results being derived from the last 5 minutes only. This is to negate the effects of any anomalies caused from the initialization of the experiment. At the end, the median value for q, k, and v are recorded as the values of the experiment. The experiment is run 100 times for several flows and radio penetration rates. Flows vary from 200 vehicles per hour to 2000 vehicles per hour. Penetration rates of 5%, 10%, 15%, 20%, 25% and 100% were considered.

4.4.2 SIMULATION RESULTS

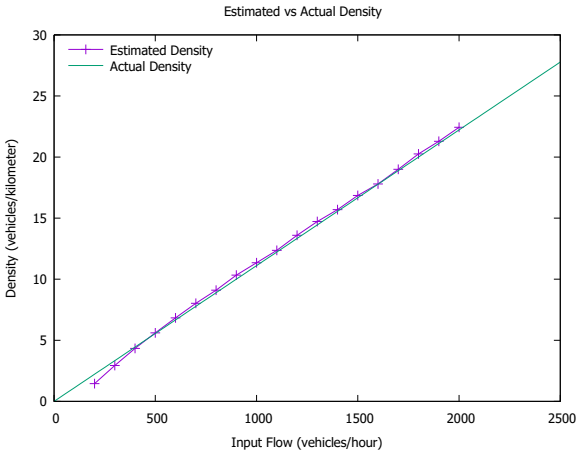
The median flow, density and velocity for each run is reported from each simulation; then averaged and plotted in the figures as shown in Figure 19.

As can be seen from Figures 19(a) and 19(b), the calculated flow and density appear indistinguishable from the actual flow and density for all values of flow above 400 vehicles per hour. While the flow and density appear indistinguishable, the calculated velocity is consistently off by 1 kilometer per hour. The graph in Figure 19(c) is specifically zoomed into show that our results are not perfect. The value for v_{avg} is given from the equation $v_{avg} = q/\kappa$, any differences in values for q and κ are reflected in the value for v. After the actual flow of 400 vehicles per hour, the flow is off by an average of less than seven vehicles per hour, and the density is off by an average of less than 0.2 vehicles per kilometer.

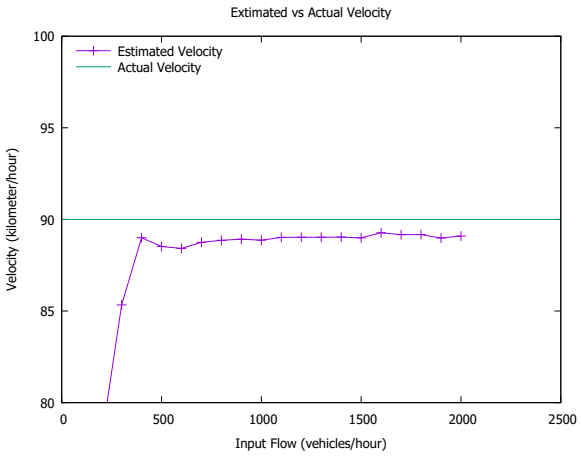
At low flows, specifically between 200 to 400 vehicles per hour, the calculated flow, density and velocity are lower than the actual result. This can be attributed to there being fewer vehicles to pass, and a likely chance that a vehicle does not pass any other between point A and point B. If the tallies are zero, this leads to the values of q and k as both being zero. For these occurrences, the value of velocity is also recorded as a zero. In the results,



(a) Mean vs. actual flow



(b) Mean vs. actual density



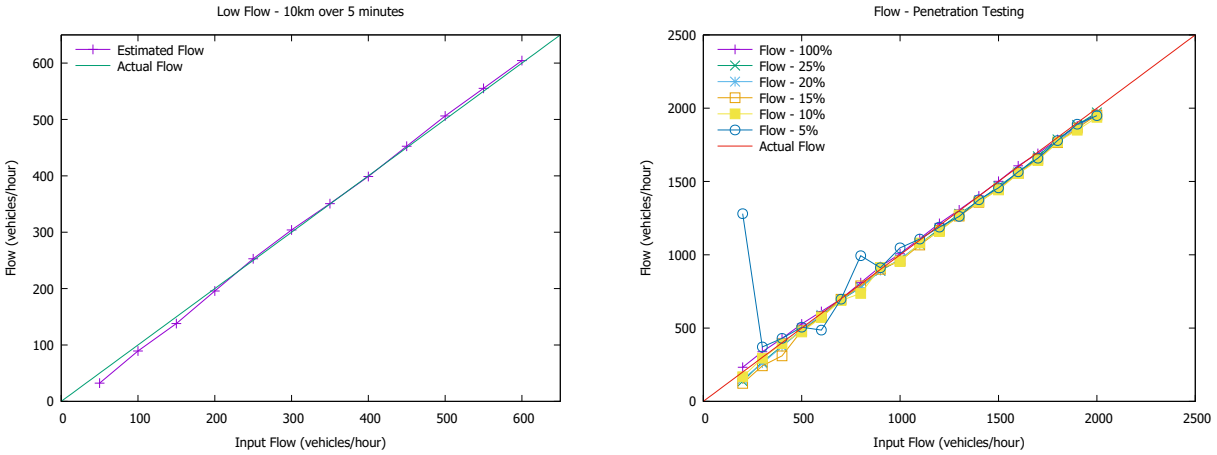
(c) Mean vs. actual velocity

Fig. 19: Mean vs. actual values for flow, density and velocity.

these values are not removed, so the plotted points for flow, density and velocity are lower for low flows. It can be expected that removing these values from the results would show closer plots for these low flows.

4.4.3 LOW FLOWS

Since the results for low flows are less than ideal, we investigated ways to increase the accuracy of the results. Specifically, we ran additional simulations for low flows starting



(a) Low flows, 10km over 5 minutes

(b) Penetration testing

Fig. 20: Mean vs. actual flow for additional experiments.

at 50 vehicles per hour to 600 vehicles per hour. The distance between point A and B is increased to 10 km and the results are measured over five minutes. As expected, the accuracy of the results increased dramatically. This can be attributed to there being more vehicles passing given the extra distance, thus more data to get better results. The results for the low flow simulations can be see in Figure 20(a).

4.4.4 PENETRATION RESULTS

Our MO method was tested for using different penetration rates of capable vehicles. In a perfect world, 100% of the vehicles on the road would be enabled with the proper hardware for this method to run. More realistically, we can only expect a certain percentage of vehicles to have the hardware required. We tested penetration rates of 5%, 10%, 15%, 20% and 25%. For the penetration tests, the rest of the parameters of the simulation remain the same as previous. The values for calculated flows are plotted in Figure 20(b).

The graph shows that the results deviate for low flows and low penetration rates, specifically for the lowest penetration rate. The calculated flows remain close to the actual flow as the flow and penetration rates increase.

With low flow and low penetration rates, the chances of enabled vehicles passing drops. It is not evident from the graphs but there are many experiments run where there are no

vehicles passing. This explains the wild swings of the 5% penetration rate plots.

4.4.5 PUTTING OUR WORK IN PERSPECTIVE

Mulligan and Nicholson [27] determined that in order for the Moving Observer method to work properly in low flows, several runs must be taken to get a good result. They determine the number of runs required to get a flow error less than 5%. Her results show that the method is only practical at a flow of 500 vehicles per hour or more over a span of 10-15km. The authors also calculated the number of trips required to reach an error rate of less than 5% by dividing the standard error, that is the standard deviation divided by square root of the number of trips, by the actual flow.

By comparison, our method improves on this previous work slightly. At 500 vehicles per hour over a span of 5km, the Mulligan and Nicholson require 8 runs, lasting a total of 24 minutes of simulation time. Our method, assuming separate 5 minute intervals as a run, only requires 4 back to back experiments, lasting a total of 20 minutes of simulation time. Using the same results, at a flow of 800 vehicles per hour, only a single experiment is required to ensure the same accuracy. As explained previously, our method does not require a specialized test vehicle, nor does it require the same vehicle turn around and observe the traffic from the opposite side of the road.

4.5 THE MO1 METHOD IN PRACTICE

In the MO1 method there is a mix of enabled and non-enabled vehicles. The method uses enabled vehicles to account for non-enabled vehicles. The method is designed to work on highways that have no additional entrances or exits. This fits well with interstate travel or long rural roads.

We now describe how the MO1 method will work using the messages and formats described in Section 3.3.

A TMC or local authority will establish a list of roadways that it requires density and flow approximations using the MO1 method. It will assign a Test Area ID to each and establish the geometry of the roadway. It will then update RSUs or local agents near the test areas with the list of test areas.

The RSU or local agent will then regularly broadcast a WSA message using the established PSID for the test area advertisement service. The WSA will include the details of the channel and details to receive WSM messages associated with test area advertisements.

The RSU or local agent will regularly broadcast the locations of nearby or upcoming

test areas so vehicles within communication range are aware of the locations. This is done through the use of the Test Area Advertisement (TAA) messages described in Section 3.3.1. The TAA message will be sent using the WSM format and will contain the Test Area Id, the MO Method Type for the test area, and the geometry describing the boundaries of the test area.

The enabled vehicles will read the messages and locate the test area geometry on its digital map. Then, using its GPS and digital map, the vehicle will determine the moment it enters into a test area.

When entering a test area, the vehicle will log the location and time of crossing into the test area; additionally, it will start a tally of all the times it is passed by codirectional vehicles or codirectional vehicles passes it. All of this information will be important when calculating density.

Upon exiting the test area, the vehicle will log its location, the time, and its current tallies. The vehicle will then determine the time spent in the test area. The vehicle will finally prepare and send a TPC message.

Remember that the TPC message is sent using WSM, meaning it may be overheard by any other enabled vehicles within communication range. This means, the vehicle may have heard TPC messages from other enabled vehicles in front of it.

The TPC message for the MO1 method includes the following details:

- Test Area Id,
- MO Method Type,
- Start Time,
- End Time,
- Density,
- Flow, and
- Tally

We note here that an alternative to this message would be to not include the density and flow in the TPC message. An advantage to sending it in the message would be for vehicles can learn the density and flow of the traffic in front of them, instead of relying solely on the infrastructure.

We now continue assuming the flow and density will be included in the TPC message. The equations for flow and density are repeated below:

$$q = \frac{l'(n_f - n_s) - l(n'_f - n'_s)}{l't - lt'}.$$

$$\kappa = \frac{t'(n_f - n_s) - t(n'_f - n'_s)}{l't - lt'}.$$

To determine the flow and density, the vehicle needs to know the following values: the length of time, t', the other vehicle spent in the test area, the tallies, $n'_f - n'_s$, and the length, l', of the test area. We note that length of the test area is known from the test area advertisement, meaning $l = l'$.

The vehicle may hear TPC message from vehicles in front of it. For each of these, the vehicle will store the values of t' and $n'_f - n'_s$. Then when it passes out of the test area, it will solve Equations (26) and (27) for each TPC message it has received. Finally it will aggregate them, for example, by averaging them together.

Having all the information required of the TPC message, the vehicle will create the message and send it to the next RSU advertising the TPC service. The vehicle will listen for WSA messages that advertise the PSID associated to the TPC service. Once it receives one, it will create the TPC message and send it to the RSU via a WSM. The TPC message includes the following fields:

- Test Area ID - This is the Test Area ID of the test area measured.
- MO Method Type - This will be 1, representing the MO1 method.
- Start Time - This is the starting time of when the vehicle entered into the test area.
- End Time - This is the ending time, when the vehicle exited the test area.
- Density - This is the aggregate density measurement as calculated by the vehicle.
- Flow - This is the aggregated flow measurement as calculated by the vehicle.
- Tally - This is the tally measurement as calculated by the vehicle.

The MO1 method is complete when the vehicle exits the test area.

4.6 SUMMARY

In this chapter we proposed a variant of Wardrop and Charlesworth's *Moving Observer* method that we have modernized and named the Mobile Observer method (MO) given that it harnesses the capabilities of the present-day vehicle. As it turns out, our variant does not suffer from the documented shortcomings of the *Moving Observer* method. The main contributing factor is that our variant considers each enabled vehicle on the road as a test vehicle, and aggregates values for flow, density, and velocity from each. Essentially, instead of a single test vehicle making several passes back and forth on the same road, several vehicles exchange tally data to gather the data at the same time, without requiring them to turn back around.

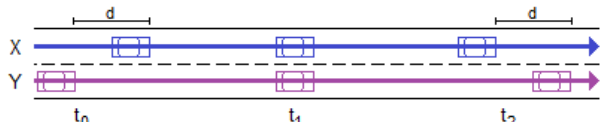
Of course, this method has some limitations. At the extreme, if there are no vehicles moving in traffic, or if there is no passing between vehicles in traffic, at any velocity, then our method will not work well. Also, in very low flows, or at very low penetration rates, where enabled vehicles do not pass often, our method is unable to deliver consistent results.

Additionally, we have shown how the method can be implemented in practice by using V2X and the message formats described in Chapter 3. The following two chapters will discuss additional methods for estimating traffic parameters based on the concept of the tally described in this chapter.

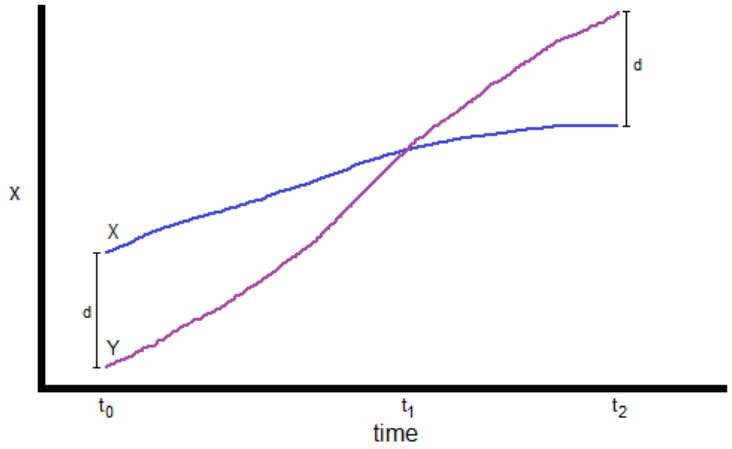
CHAPTER 5

MOBILE OBSERVER 2 METHOD

In this chapter, a second Mobile Observer method is described. The MO2 method uses the tallies of two enabled codirectional vehicles that pass one another at least once on a stretch of highway. First the method will be proven to work for a set of use cases. Then the results of simulations and sensitivity analyses is given.



(a) Roadway view



(b) Time-space view

Fig. 21: Illustrating the MO2 method.

Use Figure 21 as an illustration of the MO2 method. There are two figures representing two different views of the traffic. In Figure 21(a) shows a top down view of vehicles X and Y on the road and Figure 21(b) shows the time-space view of the two vehicles. Suppose

vehicle Y passes vehicle X at some time t_1 while in the test area. At this point in time, we assume there are zero vehicles between them. In actuality, there is a possibility that another vehicle passes at the same time, but this is considered a rare case. At the time of passing each vehicle maintains a tally and the two vehicles exchange their tallies at regular intervals. At time t_2 the two vehicles are a distance d apart from one another. They exchange tallies again and use the tallies to determine the number of vehicles between them. Since the distance between them is also known, the density can be determined.

An additional use case is where both vehicles maintain a tally at regular intervals prior to one passing the other. Given a record of past tallies, the two can determine the number of vehicles between them and densities for past times.

5.0.1 TECHNICALITIES

The concept of betweenness is important for MO2.

Consider a time interval $I = [t_1, t_2]$ with $t_1 < t_2$ and assume that Y is behind X at t_1 and that X is behind Y at t_2. Recall that $B(Y, X, t_1)$ denotes the set of vehicles between Y and X at t_1; $B(X, Y, t_2)$ denotes the set of vehicles between X and Y at t_2. Further, $b(Y, X, t_1)$ denotes the *number* of vehicles between Y and X at t_1; $b(X, Y, t_2)$ denotes the *number* of vehicles between X and Y at t_2. We are now ready to state the main technical result of this method.

Theorem 5.0.1. *Consider a time interval $I = [t_1, t_2]$ with $t_1 < t_2$ and assume that Y is behind X at t_1 and that X is behind Y at t_2. In the notation above, and assuming conservation of flow, the following relation holds:*

$$b(Y, X, t_1) + b(X, Y, t_2) = \tau(X, I) - \tau(Y, I) - 2. \tag{31}$$

Proof. We begin by evaluating the contribution of X and Y to the right-hand side (RHS) and left-hand side (LHS) of Equation (31). First, since neither X nor Y are between X and Y, their contribution to the LHS of Equation (31) is $0 + 0 = 0$. We now show that their contribution to the RHS of Equation (31) is also 0.

Indeed, recall that Y is behind X at t_1 and X is behind Y at t_2. Thus, over the time interval $I = [t_1, t_2]$, X is of Type 4 with respect to Y and Y is of Type 1 with respect to X. By Lemmas 4.2.3 and 4.2.5, combined, X and Y contribute $(+1) - (-1) = 2$ to $\tau(X, I) - \tau(Y, I)$ and, consequently, 0 to $\tau(X, I) - \tau(Y, I) - 2$, matching their contribution to the LHS.

In order to complete the proof of Theorem 5.0.1, we need the following result.

Lemma 5.0.2. *Every vehicle distinct from X and Y contributes the same amount to both $b(Y, X, t_1) + b(X, Y, t_2)$ and $\tau(X, I) - \tau(Y, I)$.*

Proof. Let Z be an arbitrary vehicle distinct from X and Y. We distinguish between the following three cases.

Case 1: Y is behind Z at t_2;

Case 2: Z is between X at Y at t_2;

Case 3: Z is behind X at t_2;

We now examine, one by one, the cases identified above.

Case 1: Y is behind Z at t_2.
This case involves the following subcases:
Subcase 1.1.: X is behind Z at t_1.
In this subcase, $Z \notin B(Y, X, t_1) \cup B(X, Y, t_2)$ and so it contributes 0 to $b(Y, X, t_1) + b(X, Y, t_2)$.

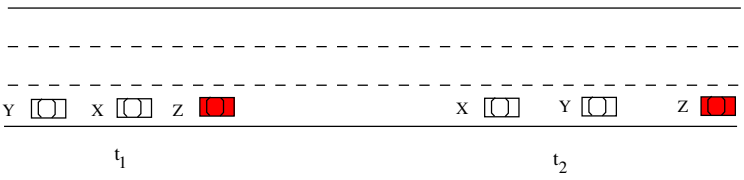


Fig. 22: Illustrating Subcase 1.1.

On the other hand, Referring to Figure 22, Z is of Type 3 with respect to both X and Y and so, by Lemma 4.2.4, it contributes $0 - 0 = 0$ to $\tau(X, I) - \tau(Y, I)$, as claimed.

Subcase 1.2.: Z is between Y and X at t_1
In this subcase, $Z \in B(Y, X, t_1)$ and $Z \notin B(X, Y, t_2)$ and, consequently, contributes $+1$ to $b(Y, X, t_1) + b(X, Y, t_2)$. At the same time, as illustrated in Figure 23, Z is of Type 1 with respect to X and of Type 3 with respect to Y. Thus, by Lemmas 4.2.3 and 4.2.4, it contributes $1 - 0 = +1$ to $\tau(X, I) - \tau(Y, I)$, also.

Subcase 1.3.: Z is behind Y at t_1

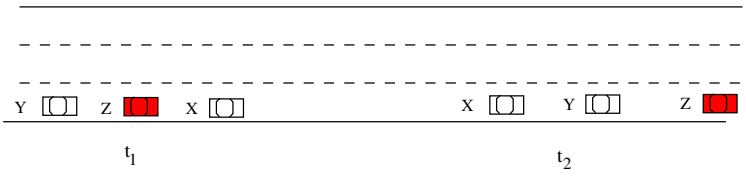


Fig. 23: Illustrating Subcase 1.2.

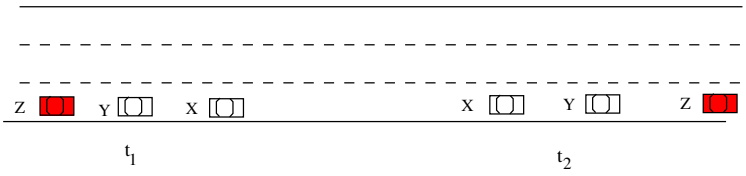


Fig. 24: Illustrating Subcase 1.3.

By the assumptions of this subcase, $Z \notin B(Y, X, t_1) \cup B(X, Y, t_2)$ and so it contributes 0 to $b(Y, X, t_1) + b(X, Y, t_2)$. Also, as can be seen from Figure 24, Z is of Type 1 with respect to both X and Y and so, by Lemma 4.2.3, it contributes $1 - 1 = 0$ to $\tau(X, I) - \tau(Y, I)$.
This completes Case 1.

Case 2: Z is between X and Y at t_2.
This case involves the following three subcases.

Subcase 2.1.: X is behind Z at t_1.

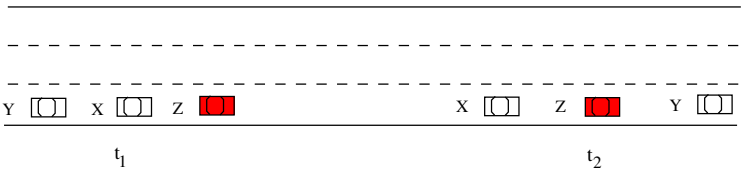


Fig. 25: Illustrating Subcase 2.1.

In this subcase, $Z \notin B(Y, X, t_1)$ but $Z \in B(X, Y, t_2)$ and so it contributes $+1$ to $b(Y, X, t_1) + b(X, Y, t_2)$.

On the other hand, referring to Figure 25, Z is of Type 3 with respect to X and of Type 4 with respect to Y; consequently, by Lemmas 4.2.3 and 4.2.5, it contributes $0 - (-1) = 1$ to $\tau(X, I) - \tau(Y, I)$, matching its contribution to $b(Y, X, t_1) + b(X, Y, t_2)$.

Subcase 2.2.: Z is between Y and X at t_1

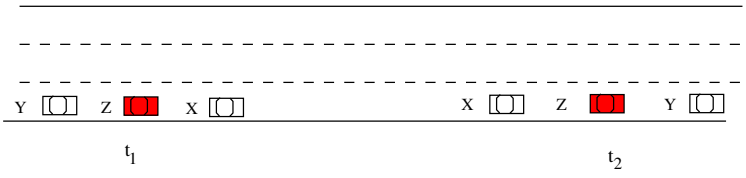


Fig. 26: Illustrating Subcase 2.2.

In this subcase, $Z \in B(Y, X, t_1) \cap B(X, Y, t_2)$ and, consequently, contributes $1 + 1 = +2$ to $b(Y, X, t_1) + b(X, Y, t_2)$. At the same time, as shown in Figure 26, Z is of Type 1 with respect to X and of Type 4 with respect to Y. Thus, by Lemmas 4.2.3 and 4.2.5, it contributes $1 - (-1) = +2$ to $\tau(X, I) - \tau(Y, I)$, as well.

Subcase 2.3.: Z is behind Y at t_1

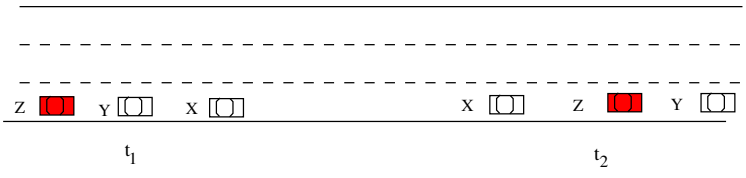


Fig. 27: Illustrating Subcase 2.3.

By the assumptions of this subcase, $Z \notin B(Y, X, t_1)$ but $Z \in B(X, Y, t_2)$ and so it contributes $0 + 1 = 1$ to $b(Y, X, t_1) + b(X, Y, t_2)$. Also, as shown in Figure 27, Z is of Type 1 with respect

to X and Type 2 with respect to Y and so, by Lemmas 4.2.3 and 4.2.4, it contributes $1 - 0 = +1$ to $\tau(X, I) - \tau(Y, I)$.

This completes Case 2.

Case 3: Z is behind X at t_2.

As before, this case involves three subcases.

Subcase 3.1.: X is behind Z at t_1.

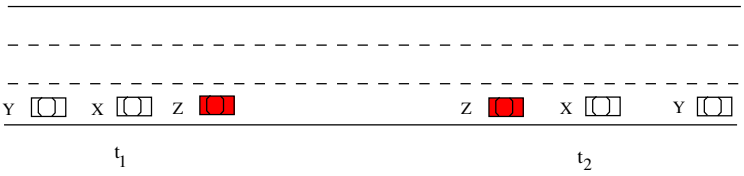


Fig. 28: Illustrating Subcase 3.1.

In this subcase, $Z \notin B(Y, X, t_1) \cup B(X, Y, t_2)$ and so it contributes 0 to $b(Y, X, t_1) + b(X, Y, t_2)$.

On the other hand, as illustrated in Figure 28, Z is of Type 4 with respect to both X and Y and so, by Lemma 4.2.5, it contributes $1 - 1 = 0$ to $\tau(X, I) - \tau(Y, I)$, matching its contribution to $b(Y, X, t_1) + b(X, Y, t_2)$, as claimed.

Subcase 3.2.: Z is between Y and X at t_1

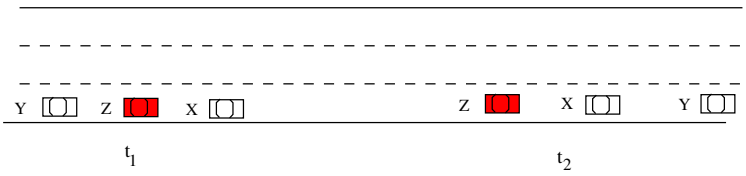


Fig. 29: Illustrating Subcase 3.2.

In this subcase, $Z \in B(Y, X, t_1)$ and $Z \notin B(X, Y, t_2)$ and, consequently, contributes $+1+0 = +1$ to $b(Y, X, t_1) + b(X, Y, t_2)$. At the same time, referring to Figure 29, Z is of Type 2 with

respect to X and of Type 4 with respect to Y. Thus, by Lemmas 4.2.4 and 4.2.5, it contributes $0-(-1)=+1$ to $\tau(X,I)-\tau(Y,I)$, also.

Subcase 3.3.: Z is behind Y at t_1

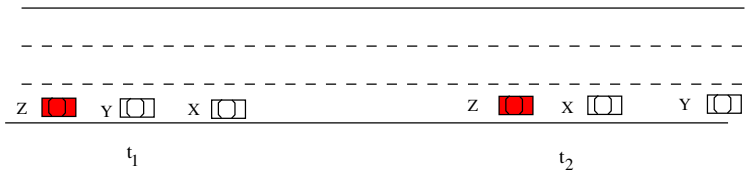


Fig. 30: Illustrating Subcase 3.3.

By the assumptions of this subcase, $Z \notin B(Y,X,t_1) \cup B(X,Y,t_2)$ and so it contributes 0 to $b(Y,X,t_1)+b(X,Y,t_2)$. Also, as shown in Figure 30, Z is of Type 2 with respect to both X and Y and so, by Lemma 4.2.4, it contributes $0-0=0$ to $\tau(X,I)-\tau(Y,I)$ as well.
This completes the proof of Lemma 5.0.2 and that of Theorem 5.0.1. □

□

5.0.2 USE-CASES

Theorem 5.0.1 is a generic result that can be instantiated in three different ways to obtain real-time roadway density estimates.

Use-case 1.
Used end-to-end, over the entire interval $I=[t_1,t_2]$, as illustrated in Figure 33, Theorem 5.0.1 allows us to estimate, from knowledge of the respective tallies $\tau(X,I)$ and $\tau(Y,I)$ collected over the interval $[t_1,t_2]$, the total number of vehicles on the roadway between Y and X at time t_1 and those between X and Y at t_2.

In this use-case, the two vehicles X and Y start communicating at time t_1 and initialize their tallies to 0. At time t_2 they exchange tally information and proceed to estimate the total number of vehicles $b(Y,X,t_1)+b(X,Y,t_2)$. For example, this use-case works well when vehicle Y is much faster than X.

Use-case 2.

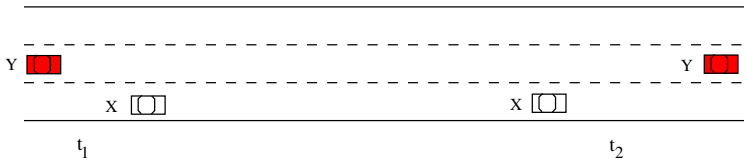


Fig. 31: Illustrating Use-case 1 for Theorem 5.0.1.

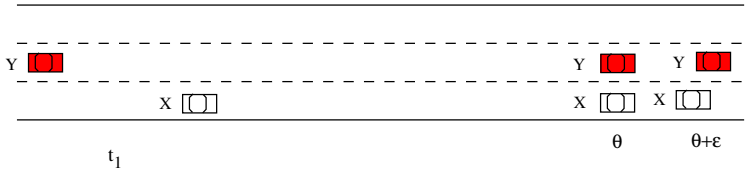


Fig. 32: Illustrating Use-case 2 for Theorem 5.0.1.

Referring to Figure 32 for an illustration, observe that since Y is behind X at time t_1 while X is behind Y at t_2, there must exist (at least) one time instant when Y passes X. We let θ, $(t_1 < \theta < t_2)$, denote the earliest such time after t_1. By Definition 2, Y is still behind X at time, θ. However, immediately afterwards, X is behind Y. More precisely, for an arbitrarily small $\epsilon > 0$, X is behind Y at time $\theta + \epsilon$. It follows that, in this use-case, Theorem 5.0.1 applies in the interval $I = [t_1, \theta + \epsilon]$.

This use-case can be applied when two vehicles X and Y, with Y behind X, establish a connection at time t_1 and initialize their tallies to 0. Later, when Y passes X for the first time, they compare tallies and, thus, obtain an estimate of $b(Y, X, t_1) + b(X, Y, \theta + \epsilon)$. In fact, in most cases $b(Y, X, \theta + \epsilon)$ will be 0, and so, for all practical purposes, what they obtain is an estimate of $b(Y, X, t_1)$, the number of vehicles *between* Y and X at time t_1. We restate Theorem 5.0.1 to reflect the conditions of this use-case.

Corollary 5.0.3. *Assume that at time t_1 vehicle Y is behind X and that Y passes X at time $\theta > t_1$. Then, for arbitrarily small $\epsilon > 0$, the following holds:*

$$b(Y, X, t_1) + b(X, Y, \theta + \epsilon) = \tau(X, I) - \tau(Y, I) - 2, \tag{32}$$

where $I = [t_1, \delta + \epsilon]$.

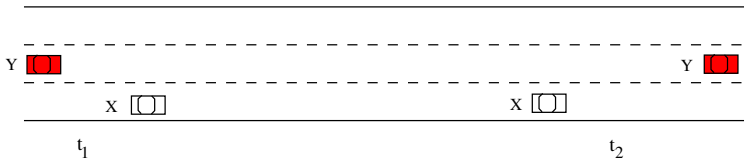


Fig. 33: Illustrating Use-case 1 for Theorem 5.0.1.

Use-case 3.

As in Use-case 2, since Y is behind X at time t_1 and X is behind Y at t_2, there must exist (at least) a time instant when Y passes X. In this use-case, we may take the latest such time θ, $(t_1 < \theta < t_2)$, before t_2. Refer to Figure **??** for an illustration. By Definition 2, Y is still behind X at the time, θ, of the passing. Thus, in this use-case, Theorem 5.0.1 applies in the interval $I = [\theta, t_2]$.

This use-case can be applied when two vehicles X and Y pass each other (without loss of generality, we may assume that Y passes X) at some time θ. At that time they initialize their tallies to 0 and then, later, at time t_2, they exchange tallies. By Theorem 5.0.1, this allows them to compute $b(Y, X, \theta) + b(X, Y, t_2)$. In fact, in most cases $b(Y, X, \theta)$ will be 0, and so, for all practical purposes, what they obtain is an estimate of $b(X, Y, t_2)$, the number of vehicles *between* X and Y at time t_2.

We restate Theorem 5.0.1 to reflect the conditions of this use-case.

Corollary 5.0.4. *Assume that at time θ vehicle Y passes X and that X is behind Y at time $t_2 > \theta$. Then, for $I = [\theta, t_2]$ the following holds:*

$$b(Y, X, \theta) + b(X, Y, t_2) = \tau(X, I) - \tau(Y, I) - 2. \tag{33}$$

To fix the ideas, from now on we will work under the assumptions of Use-case 3. For the purpose of estimating traffic density, we are interested in counting *all* vehicles that are not between X and Y at t_2 but that are "abreast" of either X or Y. More formally, a vehicle is said to be *abreast* of X or Y at t_2 if it either passes X or is passed by Y at t_2. Let us denote the set of vehicles abreast of X or Y at t_2 by $A(X, Y, t_2)$ and let $a(X, Y, t_2)$ denote their number. We are now ready to state and prove the following consequence of Corollary 5.0.4.

Corollary 5.0.5. *The number of vehicles on the roadway that are either X or Y, or are abreast of X or Y, or else are between X and Y at t_2 is*

$$\tau(X, I) - \tau(Y, I) - b(Y, X, \theta) + a(X, Y, t_2). \tag{34}$$

Proof. By Theorem 5.0.1 the number $b(X, Y, t_2)$ of vehicles *between* X and Y at t_2 is $\tau(X, I) - \tau(Y, I) - b(Y, X, \theta) - 2$. To obtain the number of vehicles in the statement of the corollary, we need to add 2 (to account for X and Y themselves) and $a(X, Y, t_2)$ to the expression above. This yields $\tau(X, I) - \tau(Y, I) - b(Y, X, \theta) + a(X, Y, t_2)$ and the proof is complete. □

The special cases where $b(Y, X, \theta) = 0$ and $a(X, Y, t_2) = 0$ are of a special practical interest. This is because the probabilities of the events that some vehicle

- is between Y and X at the very moment when Y passes X,
- passes X or is passed by Y at the exact moment where they are a distance d apart,

are exceedingly small and can be safely ignored when an *estimate* of the traffic density is sought. With this in mind, we now state the following useful corollary.

Corollary 5.0.6. *Assuming $a(X, Y, t_2) = b(Y, X, \theta) = 0$, the number of vehicles between X and Y at t_2, plus X and Y themselves, is $\tau(X, I) - \tau(Y, I)$ where $I = [\theta, t_2]$.*

Corollary 5.0.6 turns out to be the workhorse of our method to estimating traffic density on the roadway. Assume that, at time t_2 vehicles X and Y are a distance $d > 0$ apart. Observe that in Theorem 5.0.1 the distance, d, between X and Y at t_2 is a *parameter* that can be chosen in ways that make suit the needs of estimating traffic density. For example, if vehicles communicate using V2X then it turns out that a distance d close to 1 km will be advantageous.

Referring to Figure 34 assume, as above, that at time t_2, X is a distance $d > 0$ behind Y. At that time, Corollary 5.0.6 guarantees that $n = \tau(X, I) - \tau(Y, I)$ denotes the number of vehicles between X and Y, plus X and Y themselves. Let H be the random variable denoting the *headway distance* between two consecutive vehicles on the road and let $h = E[H]$ be the expectation of H; in other words, h is the mean headway distance between two consecutive vehicles.

In order to find an estimate of the traffic density on the roadway we need to determine the expected number of vehicles in a sliding window of length d. As illustrated in Figure 34, when the sliding window comprises both X and Y, the number of vehicles in the window

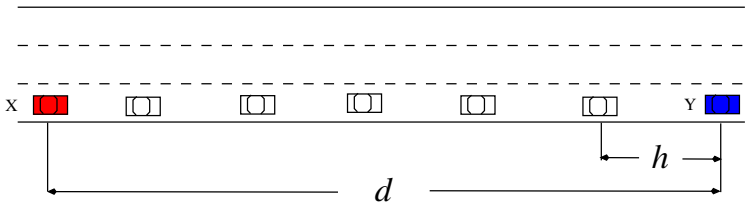


Fig. 34: Illustrating the sliding window of size d.

is n. However, if we slide the window slightly (i.e. less than h) to the left, the number of vehicles in the new window is $n-1$. We therefore need to find the weighted average of the number of vehicles in the sliding window of size d as we slide it to the left until we reach the first vehicle behind y, which by assumption is h units away from it. To make the computation meaningful, we need to evaluate the number of such windows we use.

Assuming that the length of a vehicle is l, the number of times the window is slid to the left is $\frac{h}{l}$. As mentioned before, of these sliding windows $\frac{h}{l}-1$ contain $n-1$ vehicles and one contains n.

At this point, we turn our attention to determining h as a function of d, l, and n. Indeed, since d is exactly n vehicle lengths plus $n-1$ headway distances, we can write

$$d = nl + (n-1)h.$$

Upon solving for h we obtain

$$h = \frac{d-l}{n-1} - l.$$

With the expression of h in hand, we determine the number of sliding windows as

$$\begin{aligned} \frac{h}{l} &= \frac{\frac{d-l}{n-1} - l}{l} \\ &= \frac{d-l}{l(n-1)} - 1. \end{aligned} \tag{35}$$

Next, Equation (35) allows us to determine the expected number, $N(d)$, of vehicles in a

sliding window of size d as follows

$$\begin{aligned} N(d) &= \frac{\left(\frac{d-l}{l(n-1)} - 1\right)(n-1) + n}{\frac{d-l}{l(n-1)}} \\ &= n - 1 + \frac{l(n-1)}{d-l} \\ &= (n-1)\frac{d}{d-l}. \end{aligned} \tag{36}$$

Finally, Equation (36) allows us to compute the following approximation of traffic density of the road segment of length d delimited by $x(t_2)$ and $y(t_2)$:

$$\begin{aligned} \kappa &= \frac{N(d)}{d} \\ &= \frac{(n-1)\frac{d}{d-l}}{d} \\ &= \frac{n-1}{d-l}, \end{aligned} \tag{37}$$

where, recall, $n = \tau(X, I) - \tau(Y, I)$.

5.1 MO2 SIMULATION AND RESULTS

5.1.1 MO2 SIMULATION

We tested our method with actual roadway traces provided by NGSIM I-80 dataset [124] and a simulated highway traffic dataset comprising of a 10km stretch of highway simulated using SUMO [125]. In both, the length of the road is split into two halves. As vehicles pass in the first half, they communicate and utilize Use-case 3. Upon two enabled vehicles passing, they establish themselves as a pass pair. As the moment they pass, the number of vehicles between them is assumed to be zero. They continue to communicate sharing their locations and their tallies every 100m. The density is measured using Equation (37). This value is recorded along with the time of the density measurement.

As vehicles pass in the second half, they will utilize Use-case 2. Upon two enabled vehicles establishing a connection, they exchange locations and determine the distance between themselves. Additionally, they start a tally with the assumption they may become a pass pair in the future. If and when they finally do pass, they establish themselves as a pass pair, share tallies, and determine the density between themselves. This value is recorded along with the measurement time.

5.1.2 NGSIM I-80 TRAFFIC DATA

The NGSIM I-80 dataset provides actual vehicle traces for northbound Interstate I-80, in Emeryville, California. Vehicle trace data is given every 100 milliseconds over a distance of about 500m over six lanes of traffic; our simulation will not include the traces of vehicles before the on-ramp in the dataset, cutting the distance of roadway to about 375m. The dataset is provided as two fifteen minute sets, each of which are run independently of one another. The first half, utilizing Use-case 3, and the second half, utilizing Use-case 2 are both 187.5 meters long. This dataset is known to be congested with an average density of about 350 vehicles per kilometer. See Figure 35 for a visual representation of the simulation.

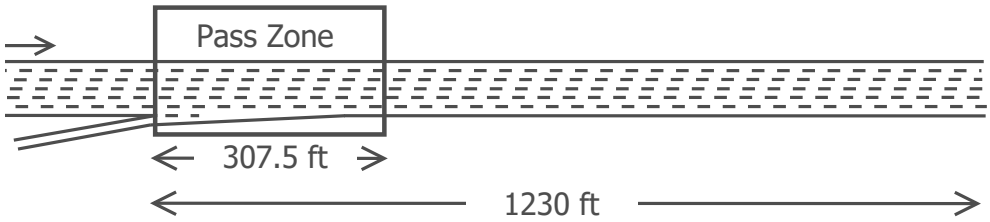


Fig. 35: NGSIM I-80 roadway

5.1.3 SIMULATED HIGHWAY TRAFFIC

In addition to NGSIM, we have simulated a stretch of highway that is not as congested. In this simulation, we have a 10km stretch of road with two lanes and no on-ramps or off-ramps. The traffic is simulated using SUMO with an average velocity of 90 km per hour. We vary the incoming flow of vehicles between 1000, 2000, and 3000 vehicles per hour. The simulation runs for three hours; however the first 30 minutes is ignored to give ample time for the roadway to initialize. The first half, utilizes Use-case 3, while the second half utilizes Use-case 2. Both halves are 5km long. Refer to Figure 5.1.3 for a visual representation of the simulation.

5.1.4 COMPARISON OF OUR METHOD

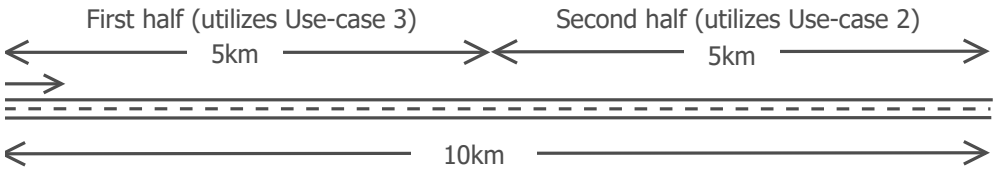


Fig. 36: Simulated highway traffic roadway

To compare the results of our method with others in literature, we looked for methods that use mobile observers under conditions similar to our own method.

One of the closest methods to our own that uses mobile observers to count passings and also experimented with different penetration rates is proposed by van Erp *et al.*[14]. They use a similar 10km stretch of highway; however, they supplement the mobile observer data with data from stationary observers. In our method, we use only mobile observer data. They report that the combined results (those including stationary and mobile observers) were better than the stationary results alone after a penetration of 2.5 - 5%.

Without a proper benchmark to compare our method to, we compared the accuracy of our estimates directly against the ground truth, as in done consistently in the literature [14, 70, 103].

5.1.5 EVALUATION DETAILS

Our experiment consists of four different datasets; there is one NGSIM dataset and three separate simulated highway datasets for flows of 1000, 2000, and 3000 vehicles per hour. Each dataset is run using four different penetration rates of enabled vehicles 5%, 10%, 25%, and 50%. Finally, each is run ten times for a total of 160 runs.

We used the simulation code as used for MO1; however, changes to accommodate the MO2 method have been added as well as code to accept NGSIM trace files.

5.1.6 SIMULATION RESULTS

In each run, the total number of vehicles are counted on the roadway each time period and divided by the length of the roadway to get the actual density. This actual density, labeled *Density Actual*, is used as a ground truth to compare the measured density.

For each time period, the minimum, maximum, 25% quartile and 75% quartile of the density are plotted using a boxplot. The measured density, labeled *Density Measured*, is the average of each measured density of the ten runs.

For NGSIM, a time period of 1 minute is chosen given the highly congested traffic. For the simulated traffic, give the traffic density is roughly an order of magnitude less, a time period of 10 minutes is chosen.

The values for each penetration rate are plotted in each set of figures. Figure 37 has each of the density plots for the NGSIM results. Figures 38, 39, and 40 have the density plots for the simulated highway traffic for flows of 1000, 2000, and 3000 vehicles per hour, respectively.

The results show a promising density estimate for high traffic flows, especially that of NGSIM, despite the very short distances of d given by this simulation approach. Additionally, the results are promising for higher penetration rates and lower traffic flows.

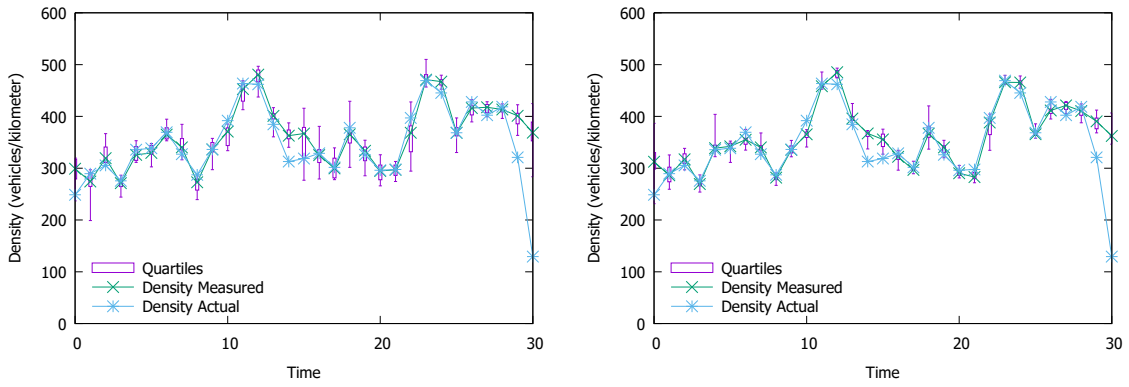
To describe the accuracy of the results, taking into account each of the runs, the relative difference is described in two ways. The relative difference between the actual density and the measured density is taken for each time period of each of the ten runs. For each time period, the mean, 75% quartile value, and the max value of each of the relative differences are taken to get the mean relative difference, the 75% relative difference, and the max relative difference. The mean relative difference and the 75% relative difference is given for the simulated traffic dataset runs; however, given the exceptional NGSIM results, the 75% relative difference and the max relative difference are given.

NGSIM results

The NGSIM results are plotted in Figure 37. Given the NGSIM datasets are provided in two independent trace files, for each set the first two minutes and last two minutes of the set are ignored in the following results; however, they are plotted in the figures.

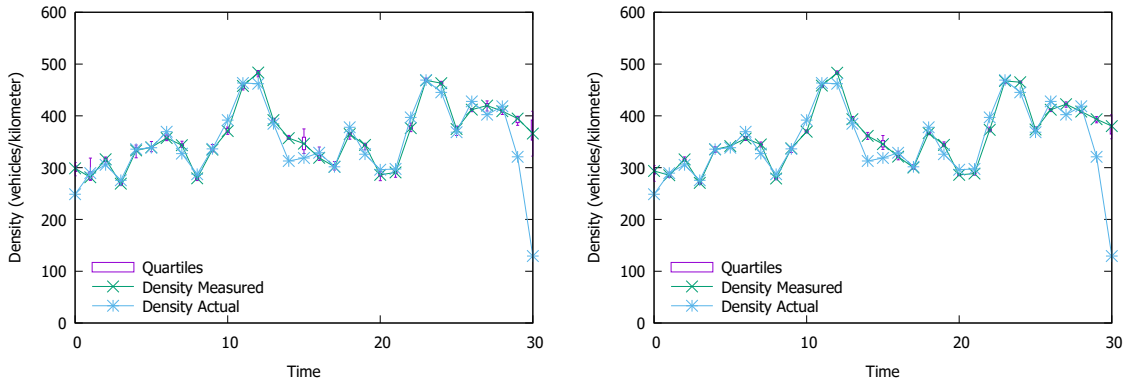
For 5% penetration rate, the 75% relative difference show excellent results, the average of the relative differences over the entire simulation is 7.2% and the maximum is 11.7%. However, with the max relative difference, it is quite clear there are outliers in the measurements; the average over the entire simulation is 12.9%, and the maximum is 20.1%.

For 10% penetration rate, the 75% relative difference rates show increased accuracy with the average of the relative differences over the entire simulation being 4.6% and the maximum being 8.2%. The max relative difference averaged over the entire simulation is 9.7% and the maximum is 20.3%.



(a) 5% Penetration, average 15.54 density readings per minute

(b) 10% Penetration, average 65.69 density readings per minute



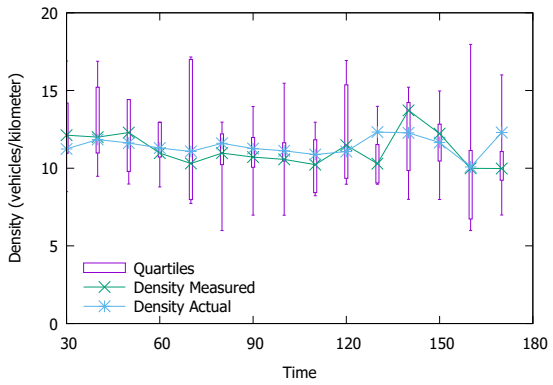
(c) 25% Penetration, average 336.40 density readings per minute

(d) 50% Penetration, average 1266.80 density readings per minute

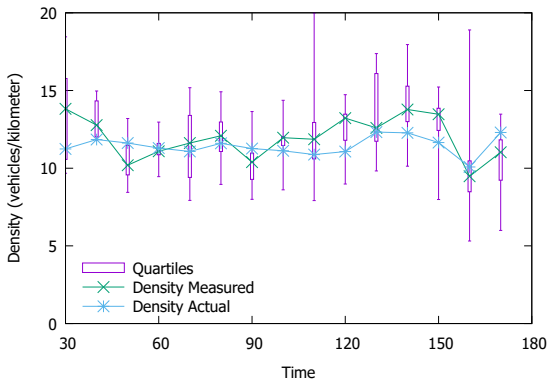
Fig. 37: Density results for NGSIM I-80.

As the penetration rate increases, each of the metrics increases with accuracy, as is evident from the plots in Figure 37. The accuracy continues to increase for the 25% penetration rate. The average of the 75% relative differences over the entire simulation is 3.4% and the maximum is only 6.0%. The average of the max relative differences over the entire simulation is 4.5% and the maximum is 7.3%.

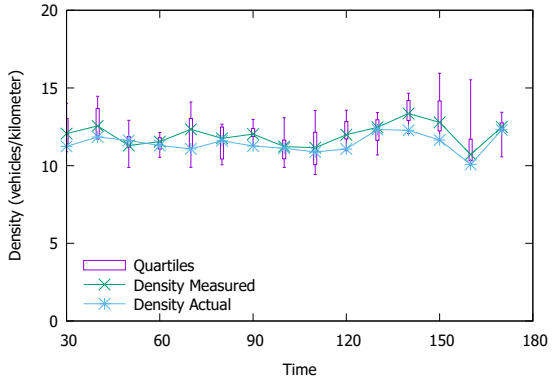
Finally, the accuracy at the 50% penetration rate is exceptional. The average of the 75% relative differences over the entire simulation is 3.1% and the maximum is 6.0%. The average of the max relative differences over the entire simulation is 3.8% and the maximum



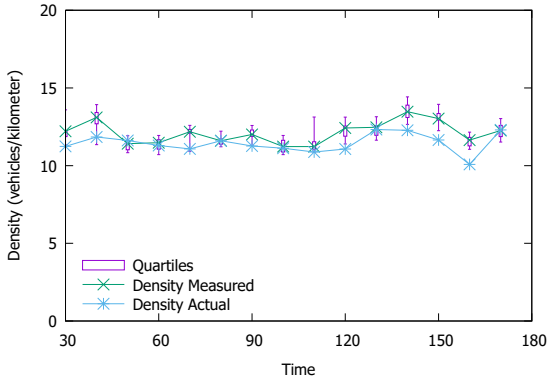
(a) 5% Penetration, average 1.63 density readings over 10 minutes



(b) 10% Penetration, average 5.43 density readings over 10 minutes



(c) 25% Penetration, average 30.17 density readings over 10 minutes



(d) 50% Penetration, average 123.13 density readings over 10 minutes

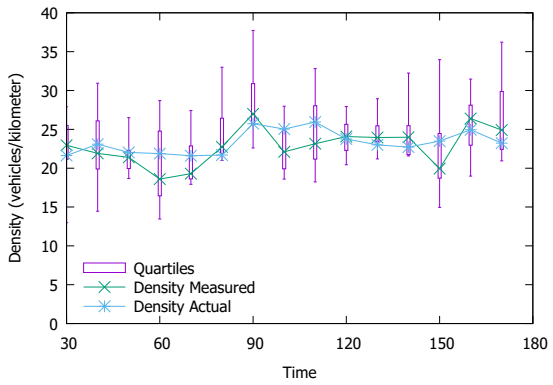
Fig. 38: Density results for simulated highway traffic, 1000 vehicles per hour.

is 6.8%.

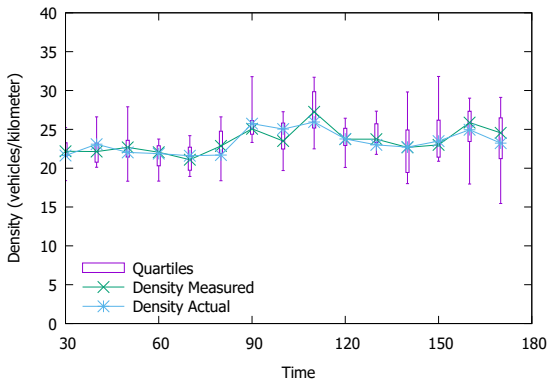
Simulated highway traffic results

The simulated traffic datasets results are split between the three flows tested. The results for the 1000, 2000, and 3000 vehicles per hour are plotted in Figures 38, 39, and 40.

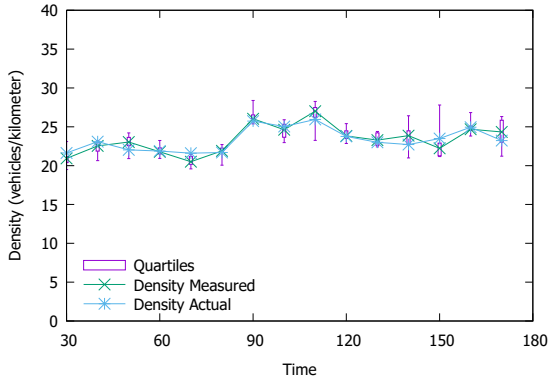
For 5% penetration rate, as expected, the mean relative difference show poor results, the average of the relative differences over the entire simulation is 18.7%, 15.2%, and 12.1% for flows of 1000, 2000, and 3000 vehicles per hour, respectively. Additionally, the 75% relative difference show equally as poor results, the average of the relative differences over the entire



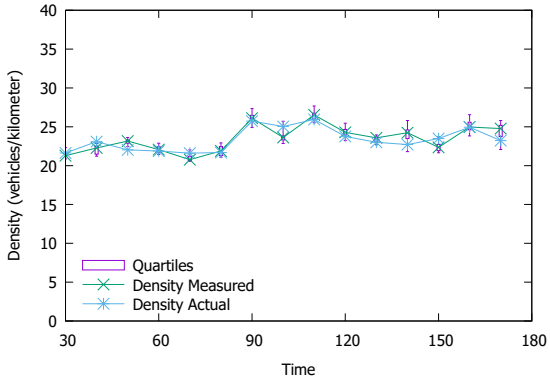
(a) 5% Penetration, average 3.72 density readings over 10 minutes



(b) 10% Penetration, average 13.73 density readings over 10 minutes



(c) 25% Penetration, average 78.433 density readings over 10 minutes



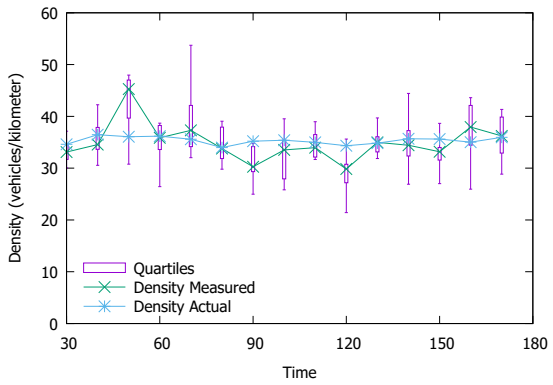
(d) 50% Penetration, average 269.89 density readings over 10 minutes

Fig. 39: Density results for simulated highway traffic, 2000 vehicles per hour.

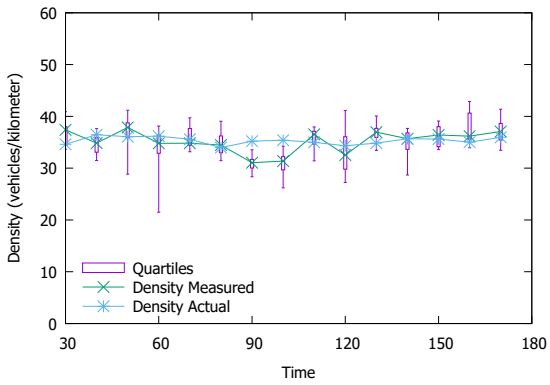
simulation is 27.5%, 22.5%, and 20.3% for flows of 1000, 2000, and 3000 vehicles per hour, respectively.

For 10% penetration rate, the mean relative difference is better with the average of the relative differences over the entire simulation being 14.9%, 11.1%, and 8.7% for flows of 1000, 2000, and 3000 vehicles per hour, respectively. Additionally, the 75% relative difference rates show increased accuracy with the average of the relative differences over the entire simulation being 23.8%, 17.6%, and 14.9% for flows of 1000, 2000, and 3000 vehicles per hour, respectively.

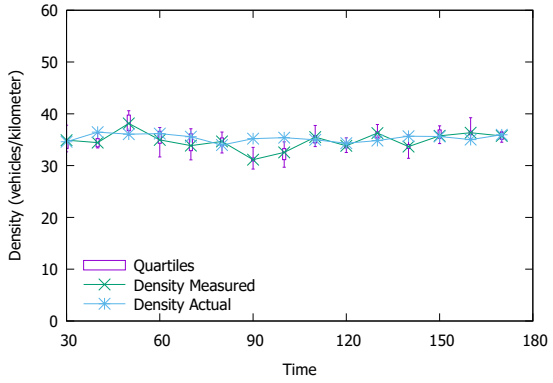
As the penetration rate increases, each of the metrics increases with accuracy, as is



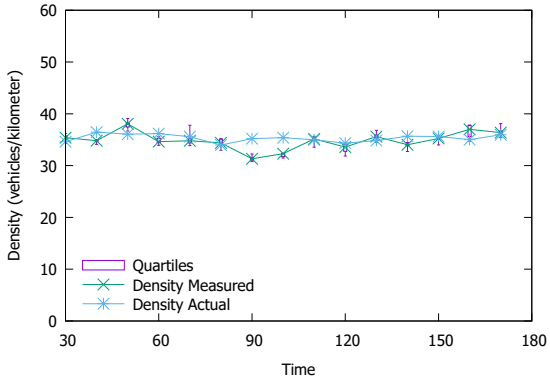
(a) 5% Penetration, average 6.79 density readings over 10 minutes



(b) 10% Penetration, average 22.80 density readings over 10 minutes



(c) 25% Penetration, average 123.02 density readings over 10 minutes



(d) 50% Penetration, average 472.85 density readings over 10 minutes

Fig. 40: Density results for simulated highway traffic, 3000 vehicles per hour

evident from the plots. The accuracy continues to increase for the 25% penetration rate. The mean relative difference show excellent results; the average of the relative differences over the entire simulation is 7.0%, 9.2%, and 6.7% for flows of 1000, 2000, and 3000 vehicles per hour, respectively. Additionally, the average of the 75% relative differences over the entire simulation is 12.4%, 12.3%, and 9.3% for flows of 1000, 2000, and 3000 vehicles per hour, respectively.

Finally, the accuracy at the 50% penetration rate is exceptional; for the mean relative difference, the average of the relative differences over the entire simulation is 6.9%, 8.3%, and 5.6% for flows of 1000, 2000, and 3000 vehicles per hour, respectively. Additionally, The

average of the 75% relative differences over the entire simulation 9.1%, 10.0%, and 7.2% for flows of 1000, 2000, and 3000 vehicles per hour, respectively.

5.2 SENSITIVITY ANALYSES

The first goal of this section is to present the details of the algorithm for tally exchange. The second goal is to offer sensitivity analysis results of our method. Specifically, Section 5.2.1 discusses sensitivity results of our method to the aggregation distance. Next, Section 5.2.2 shows how the accuracy of our method depends on penetration rate.

Throughout this section, we assume that vehicles engage in V2X-compliant V2V communications with a maximum transmission range $t_x = 1000$m.

5.2.1 SENSITIVITY OF OUR METHOD TO TALLY EXCHANGE DISTANCE

This subsection reports on our results regarding the sensitivity of our method to the tally exchange distance. To measure the sensitivity of the tally exchange distance, we have run simulations over a 10km stretch of roadway using flow rates of 1000, 2000, and 3000 vehicles per hour. The penetration rate is set to 25% and each simulation is run 10 times.

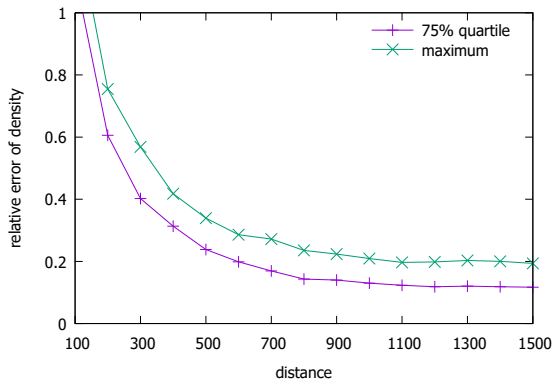
The tally exchange distance, d, that is the distance between the trailing and leading vehicles, is varied between 100m and 1500m. For each tally exchange distance, the densities are aggregated every minute. The 75% quartile and the maximum of each minute is recorded for each tally exchange distance and plotted in Figures 41(a), 41(b), and 41(c) for flows of 1000, 2000, and 3000 vehicles per hour, respectively.

The results show that the relative error of the density for each of the flows begin to stabilize at 800m. It should be noted that in Figure 41(c), the results begin to degrade at 1200m. This is due to a decreased number of vehicles reaching a tally exchange distance of 1200m or more given the high flow rate and increased congestion.

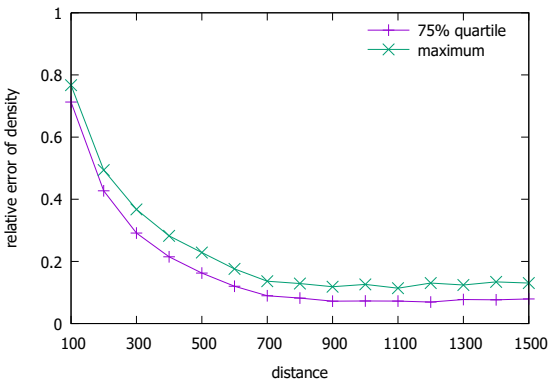
5.2.2 SENSITIVITY OF OUR METHOD TO PENETRATION RATE

In this subsection we offer a sensitivity analysis of our method to penetration rate. We use the same simulation setup as in Section 5.2.1 except that we also vary the penetration rate using values 5%, 10%, 25%, and 50%.

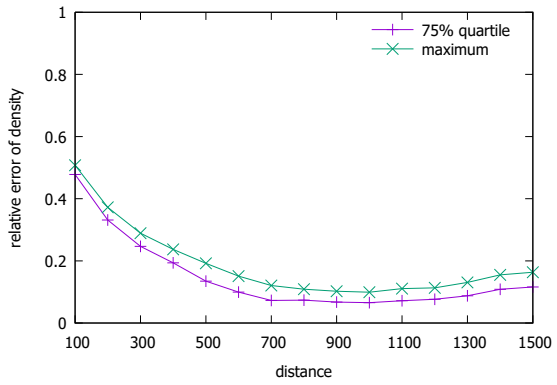
To measure the sensitivity to penetration rate, we have aggregated each of the density measurements for a ten minute time span by averaging them in the obvious way. As the penetration rate increases, the range of measured values over each of the ten runs converges.



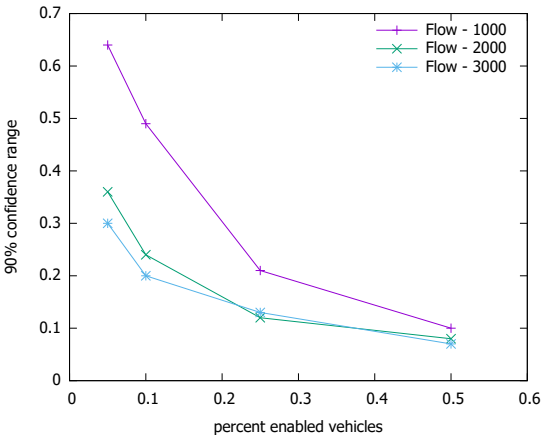
(a) *Flow of 1000 vehicles per hour*



(b) *Flow of 2000 vehicles per hour*



(c) *Flow of 3000 vehicles per hour*



(d) *90% confidence range over penetration rates*

Fig. 41: Relative density error over distance

We made the assumption that the distribution of aggregated density values for each of the ten runs is normally distributed. We computed the standard deviation and multiplied it by 1.65 to give a range for the mean value that represents 90% of the expected density values. Using this, we can determine the range which represents 90% of the expected density values. The values are plotted in Figure 41(d). For example, the plot for the flow of 3000 vehicles can be read as follows: given a penetration rate of 5%, the density measurement is expected to be within 30% of the mean with 90% confidence.

At a penetration rate of 25%, the density measurement for a flow of 1000 vehicles is expected to be within 21% of the mean with 90% confidence. For a flow of 2000 vehicles,

12%, and for 3000 vehicles, 13%.

5.2.3 COMPARISON WITH MO1

The MO1 method is unique in that it measures the flow and density at the same time. Also, as opposed to MO2, it does not require its parameters to be shared with a vehicle it passes.

This means, in the MO1 method, the parameters can be shared with more vehicles than the MO2 method. Because of this the MO1 method is expected to be more accurate than MO2 in very low flow or low penetration rate scenarios. However, if the traffic flow varies widely, this is expected to result in less accurate results.

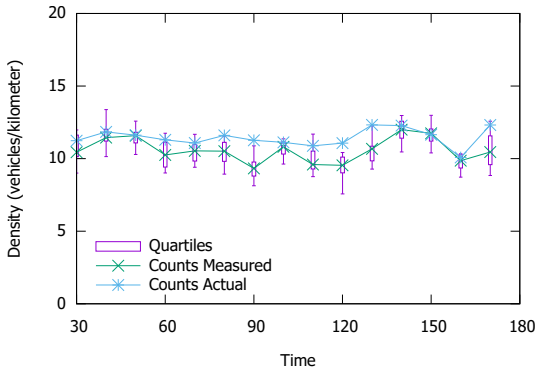
The MO2 method only measures the density of vehicles and does require the MO partners pass within the test area for the method to work. Because the method counts the exact density of the two MO partners, the MO2 method will collect better instantaneous density measurements. As variability in the traffic flow increases, the MO2 method is expected to produce better results than the MO1 method. Additionally, as the penetration rate of vehicles increases, it is expected the MO2 method will produce better results.

To directly compare the MO1 and MO2 methods, another MO1 simulation was run using the same test scenarios as MO2. The same simulated highway traffic simulation, described in Section 5.1.3, as run for MO2 was run for MO1. The MO1 results for flows of 1000, 2000, and 3000 are shown in Figures 42(a), 42(c), and 42(e), respectively. The MO2 results for flows of 1000, 2000, and 3000 are shown in Figures 42(b), 42(d), and 42(f), respectively. The MO1 and MO2 methods both produce results with similar accuracy of density in all three of the scenarios. The only benefit of the MO1 method is that it also produces highly accurate flow measurements as well.

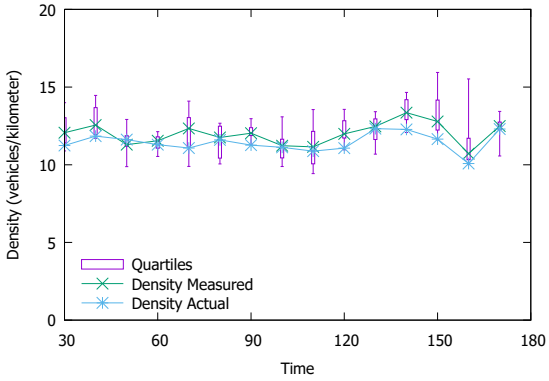
5.3 THE MO2 METHOD IN PRACTICE

In the MO2 method there is a mix of enabled and non-enabled vehicles. The method uses enabled vehicles to account for non-enabled vehicles. Additionally, an enabled vehicle must have the ability to communicate wirelessly with other enabled vehicles and establish an agreement to run the MO2 method. This method is designed to work on long stretches of road that have no additional entrances or exits. This fits well with interstate travel or long rural roads.

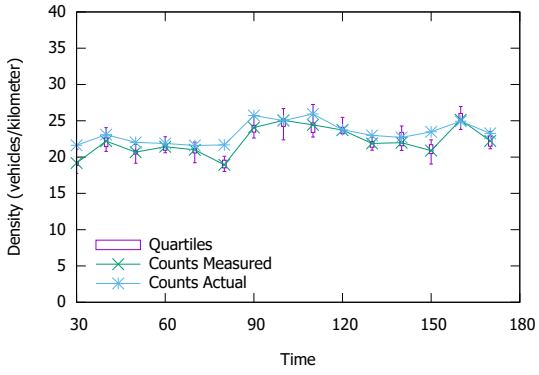
We now describe how the MO2 method will work using the messages and formats described in Section 3.3.



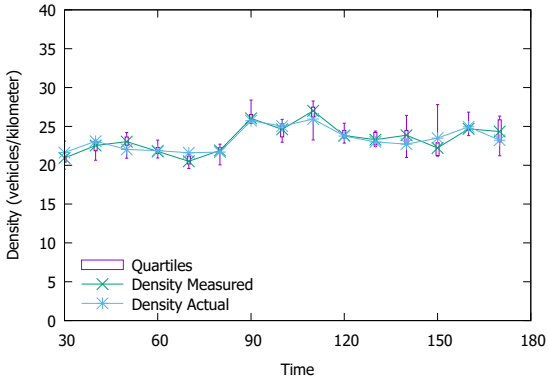
(a) MO1 SUMO 1000 vehicles per hour



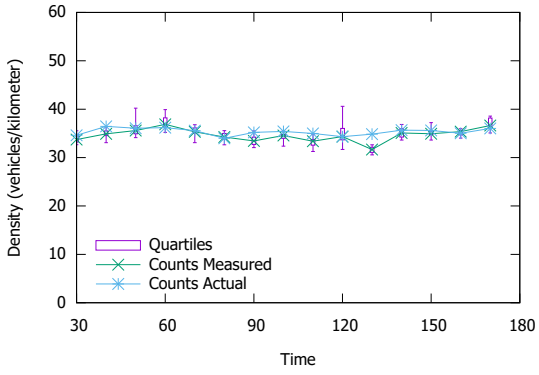
(b) MO2 SUMO 1000 vehicles per hour



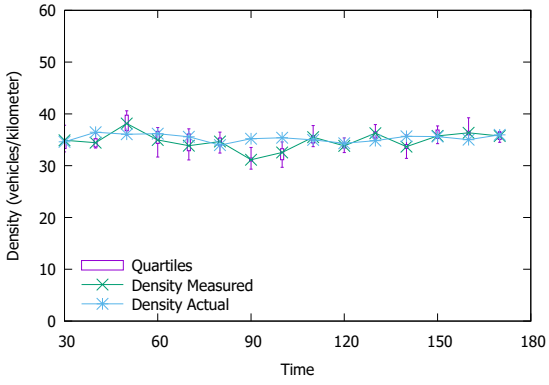
(c) MO1 SUMO 2000 vehicles per hour



(d) MO2 SUMO 2000 vehicles per hour



(e) MO1 SUMO 3000 vehicles per hour



(f) MO2 SUMO 3000 vehicles per hour

Fig. 42: Comparison of MO1 and MO2 density results.

A TMC or local authority will establish a list of roadways that it requires density approximations using the MO2 method. It will assign a Test Area ID to each and establish the geometry of the roadway. It will then update RSUs or local agents near the test areas with the list of test areas.

The RSU or local agent will then regularly broadcast a WSA message using the established PSID for the test area advertisement service. The WSA will include details of the channel and details to receive WSM messages associated with test area advertisements.

The RSU or local agent will regularly broadcast the locations of nearby or upcoming test areas so vehicles within communication range are aware of the locations. This is done through the use of the Test Area Advertisement (TAA) messages described in Section 3.3.1. The TAA message will be sent using the WSM format and will contain the Test Area Id, the MO Method Type for the test area, and the geometry describing the boundaries of the test area.

The enabled vehicles will read the messages and locate the test area geometry on its digital map. Then, using its GPS and digital map, the vehicle will determine the moment it enters into a test area.

Upon entering a test area, the vehicle will advertise itself as an enabled vehicle. It will start the MO Partner Negotiation by sending a MO Partner Request (MPR) message, as described in Section 3.3.3. This message is a broadcast message sent using UPD over IPV6. This message will include the Test Area Id, the MO Method Type, and its own Temporary Id. Note that the Temporary ID used in the MPR message is the same that is used in the BSM message; this ensures any vehicles receiving the MPR can determine the location of the sender. The MPR message is sent regularly throughout the test area to ensure it can establish a MO Partner with any possible enabled vehicles.

When an enabled vehicle receives an MPR, it will check if it is within the same Test Area that is send in the MPR message. If it is, it will then determine the location, velocity, and heading of the sender from a BSM message with the same Temporary ID as the MPR message. If both are traveling in the same direction it will to respond the MPR with a MO Partner Accept (MPA) message. This message is described in Section 3.3.4. The vehicle receiving the MPR message knows the IPV6 address of the sender from the From Address of the UDP message. When responding it will send the MPR message using TCP over IPV6 and will send it to this same From Address. This message will include the Test Area Id, Start Time, MO Method Type, and its own Temporary Id. The receipt and acknowledgment of the MPR message establishes that they are both part of an MO Partnership which starts at

the Start Time. This Start Time will be the time of the responding vehicles MPR message.

During the MO Partner Negotiation, each vehicle sends to the other its Temporary Id. Given that BSM messages are send regularly, each MO Partner will track the location, velocity, and heading of the other partner. Specifically, each MO Partner will store the IPV6 address, Temporary Id, and Start Time. For the first BSM received by its partner, the vehicle will determine which of the two is the leading and which is the trailing vehicle. Each time it receives a BSM from the Temporary ID of its partner it will update the location and the distance between the two of them. Finally, it will maintain a tally of all the times it is passed by codirectional vehicles or codirectional vehicles passes it. All of this information will be important when calculating density.

Each partner will track the other using the BSM messages. Each will regularly estimate the time they will pass. Note that this does not need to be done with every BSM message received; however, as the two get closer and closer, it will need to estimate the time they pass more and more to establish the exact moment they pass.

Upon the moment they pass, each will create a Tally Exchange Message (TEM) as described in Section 3.3.5. The TEM will contain the data required to determine the density at the time the MO partnership was established using Equation (37) which is replicated below:

$$\kappa = \frac{n-1}{d-l},$$

The required information to determine the density at the start time is:

- the value for n, given by the tally of the vehicle in front minus the tally of the vehicle behind,
- the distance, d, between the two vehicles,
- the average length of a vehicle, l, which may be a hardcoded value.

The TEM will include the following fields which are used by both vehicles to share the above data points so each can have an approximation for the density at the start time:

- Test Area Id, which will be used to verify the test area being measured.
- Start Time - This is the start time as sent in the MPR message.

- End Time - This is the time that the vehicle determined as the time of passing. This value should be the same for both vehicles. If the two vehicles disagree on the End Time, there may be a need to reconcile the difference. This reconciliation is not in scope for this work.
- Start Location - This is the starting location of the vehicle.
- End Location - This is the final location of the vehicle. Both vehicles should have similar values; if they differ by a large amount there may be a need to reconcile the difference. Again, this reconciliation is not in scope for this work.
- Heading - This represents the heading of the vehicle.
- Tally Type - This field is left blank for the MO2 method.
- Tally Codirectional - This is the tally of vehicle, $n_f - n_s$.
- Tally Oncoming - This field is left blank for the MO2 method.
- Flow - This field is left blank for the MO2 method.
- Density - This field is left blank for the MO2 method.

Each vehicle will send its TEM message using TCP over IPV6 with the respective IPV6 To and From address fields filled appropriately. After retrieving the message from its MO partner, each will determine the density between them at the start time using Equation (37).

Since each has an estimated density, both will create a Traffic Parameter Collection (TPC) message to send to the next RSU advertising the TPC service. The vehicle will listen for WSA messages that advertise the PSID associated to the TPC service. Once it receives one, it will create the TPC message and send it to the RSU via a WSM. The TPC message includes the following fields:

- Test Area ID - This is used to verify the test area being measured.
- MO Method Type - the MO Method Type serves two purposes. First it is used to verify the method used, and second, it is used to determine the format of the MO Measurement Data.

- Start Location - The Start Location is the first field of the MO Measurement Data. It includes the location data of the trailing vehicle at the time of the density measurement.
- End Location - The End Location field is the second field of the MO Measurement Data. It includes the location data of the leading vehicle at the time of the density measurement.
- Time - The Time field is the third field of the MO Measurement Data. It is the time of the density measurement.
- Density - The Density field is the fourth and final field of the MO Measurement Data. It is the density read between the Start Location and End Location at the time represented in the Time field.

Also, during this time, the old tallies from between the start time and the time they passed will be saved. New tallies, starting at the time they both passed, will be started. Each vehicle will continue to track the location of one another using BSM messages and will determine the distance between them. Upon reaching a distance d apart they will both send another TEM message. This time the Start Time and Start Location will be the time and location where they passed. The End Time and End Location will be the current time and the vehicles current location. Instead of a single distance, d, there may be a set $d_1, d_2 ... d_n$ of n distances where they will exchange TEM messages. This will provide a change in the density over time and may be of interest to the TMC.

Each time a density estimate is determined, the vehicle will create a new TPC message to be sent to the next RSU that is advertising the Traffic Parameter Collection Service.

While in the Test Area, the vehicles in an MO Partnership will not change their Temporary Id. Each vehicle in the MO Partnership will track the other, using the temporary Id, location, velocity, and heading from the BSM messages. Note if either vehicle changes its Temporary Id, its MO partner can no longer track it.

The MO2 method is complete when:

- the two MO Partners stop getting messages from the other, or
- the leading vehicle leaves the test area.

5.4 SUMMARY

The main contribution of the MO2 is to propose a novel method for real-time estimation of highway traffic density. The main advantages of this traffic density estimation method are its simplicity, the fact that it is lightweight in terms of the computational resources needed, and that it only needs V2V communications to aggregate density estimates and to disseminate them to fellow drivers. In addition, it turns out that our method is also privacy-preserving, a definite advantage.

What sets the method apart is that while the vast majority of the density estimation methods known to us require aggregating probe vehicle data with data obtained from various flavors of stationary sensors, our method is self-sufficient in the sense that such aggregations are not necessary. Extensive simulations were performed using actual vehicle traces using the NGSIM I-80 dataset along with SUMO-generated synthetic traffic traces. These simulation results have shown that our method remains accurate even at low penetration rates.

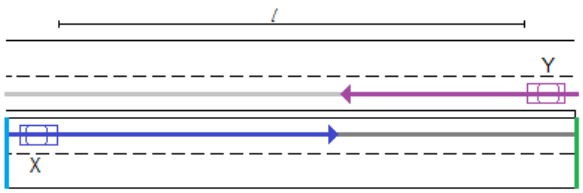
In comparison to the MO1 method, MO2 uses the same idea of the tally; however, here we exploit an assumption that the number of vehicles between two passing enabled vehicles is zero. In the next chapter we continue to exploit this assumption and use it to estimate the density and flow in urban traffic.

CHAPTER 6

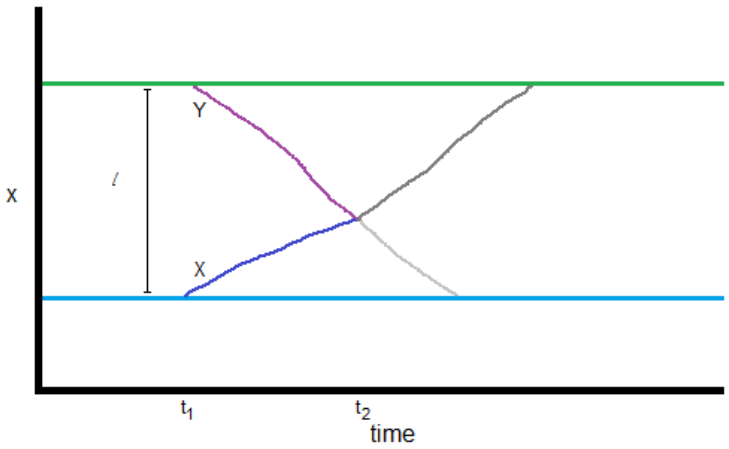
MOBILE OBSERVER 3 METHOD

In the previous chapter, we looked at the MO2 method where the assumption that there are likely zero vehicles between two passing codirectional vehicles is exploited. Given the relative difference of velocities between codirectional vehicles is quite small, the method required a long stretch of road to work.

In this chapter, we consider two partner vehicles in oncoming traffic. The relative difference of their velocities will be large, and as such, the stretch of road can be small. Specifically, we look at this method in the context of an urban setting, like the stretch of roadway between two intersections.



(a) Roadway view



(b) Time-space view

Fig. 43: Illustrating the MO3 method.

First the method will be proven to work and then the results of simulations and sensitivity analyses are given. Due to an unexpected result, another MO3-Flow method is also described, proven to work, and the results of simulations and sensitivity analyses are given.

Again, as with MO2, in MO3 the concept of betweenness is very important. The tallies are again used to determine the number of vehicles between them.

Use Figure 43 as an illustration of the MO3 method. There are two figures representing two different views of the traffic. In Figure 43(a) shows a top down view of vehicles X and Y on the road and Figure 43(b) shows the time-space view of the vehicles. Consider a vehicle X traveling in the direction of interest. As it enters into the test area at t_1 it will maintain a tally of codirectional traffic. An oncoming vehicle Y enters into the test area also at time t_1 and maintains a tally of oncoming traffic. Upon both vehicles meeting at time t_2, they exchange tallies. These tallies are used to determine the number of vehicles between them at time t_1 and the density of the roadway between them at time t_1.

6.0.1 PRELIMINARIES

Definition 9. *Given vehicles X and Y moving in* opposite *directions, we say that X* meets *Y at time t, $(t > 0)$, if $x(t) = y(t)$ and there exists $\delta > 0$ such that for all ϵ, $(0 < \epsilon \leq \delta)$, $x(t - \epsilon) < y(t - \epsilon)$.*

Refer to Figure 44 for an illustration. Observer that X meets Y at time t.

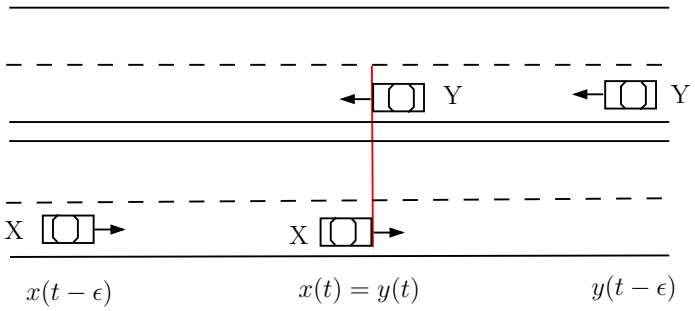


Fig. 44: X meets Y at time t.

We note that, at time t, several vehicles can meet vehicle Y moving in the opposite direction. Obviously, the number of such vehicles is upper-bounded by the number of traffic lanes in the opposite direction of Y.

Consider a time interval $I = [t_1, t_2]$ with $t_1 < t_2$ and vehicles X and Y moving in opposite directions. We assume that at time t_1, $x(t_1) < y(t_1)$ and that, subsequently, X meets Y at time t_2.

We find it useful to categorize the vehicles *codirectional* with X in terms of their relative position with respect to X and Y in the time interval I.

Definition 10. *Let $\overline{\tau}(Y, I)$ be the number of* oncoming *vehicles that have met Y in the time interval I. At the risk of mild confusion, we refer to $\overline{\tau}(Y, I)$ as the* tally *of Y in I.*

One of the main contributions of this work is to show that the density between X and Y can be computed in terms of the tallies $\tau(X, I)$ and $\overline{\tau}(Y, I)$ of X and Y. In order to state our main result, we need a number of intermediate results that will be developed in Section 6.0.2.

6.0.2 TECHNICALITIES

As it turns out, the tally concept in Definitions 8 and 10 is fundamental to approximating traffic density. We begin by stating and proving a number of technical results.

First, Lemmas 4.2.3, 4.2.4, and 4.2.5 are restated, as they are used in this section.

Lemma 4.2.3 *Every vehicle of Type 1 with respect to X in I contributes $+1$ to $\tau(X, I)$.*

Lemma 4.2.4 *Every vehicle of Type 2 or Type 3 with respect to X in I contributes 0 to $\tau(X, I)$.*

Lemma 4.2.5 *Every vehicle of Type 4 with respect to X in I contributes -1 to $\tau(X, I)$.*

We now extend these results to the oncoming tallies.

Lemma 6.0.1. *Every vehicle of Type 1 with respect to X in I contributes $+1$ to $\overline{\tau}(Y, I)$.*

Proof. Consider an arbitrary vehicle C of Type 1. By definition, X is behind C at t_2 and, consequently, C meets Y before X does. It follows that C contributes $+1$ to $\overline{\tau}(Y, I)$. □

Lemma 6.0.2. *Every vehicle of Type 2 with respect to X in I contributes 0 to $\overline{\tau}(Y, I)$.*

Proof. Consider an arbitrary vehicle C of Type 2. By definition, C is behind X at time t_2 and, as a result, C does not meet Y in I. It follows that C contributes 0 to $\overline{\tau}(Y, I)$. □

Lemma 6.0.3. *Every vehicle of Type 3 with respect to X in I contributes* $+1$ *to* $\overline{\tau}(Y, I)$.

Proof. Consider an arbitrary vehicle C of Type 3. By definition, X is behind C at t_2 and, consequently, C meets Y before X does. It follows that C contributes $+1$ to $\overline{\tau}(Y, I)$. □

Lemma 6.0.4. *Every vehicle of Type 4 with respect to X in I contributes* 0 *to* $\overline{\tau}(Y, I)$.

Proof. Consider an arbitrary vehicle C of Type 4. By definition, C is behind X at time t_2 and, as a result, C does not meet Y in I. It follows that C contributes 0 to $\overline{\tau}(Y, I)$. □

Consider, again, the time interval $I = [t_1, t_2]$ with $t_1 < t_2$ and vehicles X and Y moving in opposite directions. We assume without loss of generality that, at time t_1, $y(t_1) - x(t_1) = L$. In this work we are interested in approximating, at time t_1, the density $\kappa(X, Y, t_1)$, of the traffic codirectional with X, delimited by the road segment with coordinates $[x(t_1), y(t_1)]$ and length L. Any algorithm for approximating traffic density has to count vehicles. In our algorithm, a generic vehicle Z, codirectional with X, is counted and contributes to $\kappa(X, Y, t_1)$ if and only if, X is behind Z at t_1 and

$$x(t_1) \leq z(t_1) \leq y(t_1). \tag{38}$$

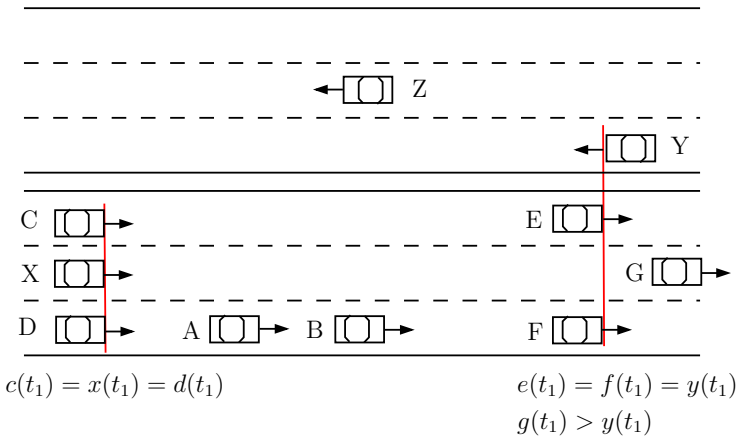


Fig. 45: Illustrating the density, at t_1, of the road segment delimited by X and Y.

Referring to Figure 45, X is behind A, B, E, and F. In addition, $a(t_1) < b(t_1) < y(t_1)$ and $e(t_1) = f(t_1) = y(t_1)$ and so all these vehicles will be counted. Notice that X is passing D at t_1 and so, by Definition 1, X is behind D; since $d(t_1) = x(t_1) < y(t_1)$, D will be counted. However, C is passing X at t_1. By Definition 1, C is *behind* X at t_1, even though $c(t_1) = x(t_1)$. Thus, C will be excluded from the computation of the density. Likewise, G is excluded because $g(t_1) > y(t_1)$. Obviously, vehicles in the oncoming direction are not counted: thus, in the figure Z is not included in the computation of density. Thus, in Figure 45 six vehicles, X, D, A, B, E and F will be counted, and the density at t_1 of the road segment delimited by $x(t_1)$ and $y(t_1)$ is approximated by $\frac{6}{L}$.

We are now in a position to state and prove the main result of this section.

Theorem 6.0.5. *Consider a time interval $I = [t_1, t_2]$ with $t_1 < t_2$ and vehicles X and Y traveling in opposite direction with $y(t_1) - x(t_1) = L > 0$. Assuming that X meets Y at a later time t_2, the density $\kappa(X, Y, t_1)$ of the road segment between X and Y at t_1 is*

$$\kappa(X, Y, t_1) = \frac{\overline{\tau}(Y, I) - \tau(X, I)}{L}. \tag{39}$$

Proof. We are interested in approximating the density of the road segment between X and Y at t_1. Recall that a vehicle Z contributes to $\kappa(X, Y, t_1)$ if and only if X is behind Z at t_1 and, in addition, Equation (38) above is satisfied.

By Definitions 7 and 8, the only vehicles that satisfy Equation (38) are those of Types 3 and 4. Consequently, in order to prove the theorem, we need to show that every vehicle of Type 3 and 4 contributes $+1$ to $\overline{\tau}(Y, I) - \tau(X, I)$, while all other vehicles contribute 0.

First, consider an arbitrary vehicle of Type 1. By Lemmas 4.2.3 and 6.0.1 this vehicle contributes $+1$ to both $\tau(X, I)$ and $\overline{\tau}(Y, I)$ and so contributes 0 to $\overline{\tau}(Y, I) - \tau(X, I)$.

Next, consider an arbitrary vehicle of Type 2. By Lemmas 4.2.4 and 6.0.2 this vehicle contributes 0 to both $\tau(X, I)$ and $\overline{\tau}(Y, I)$ and so contributes 0 to $\overline{\tau}(Y, I) - \tau(X, I)$.

Further, consider an arbitrary vehicle of Type 3. By Lemmas 4.2.4 and 6.0.3 this vehicle contributes 0 to $\tau(X, I)$ and $+1$ to $\overline{\tau}(Y, I)$ and so contributes $+1$ to $\overline{\tau}(Y, I) - \tau(X, I)$.

Finally, consider an arbitrary vehicle of Type 4. By Lemmas 4.2.5 and 6.0.4 this vehicle contributes 0 to $\tau(X, I)$ and -1 to $\overline{\tau}(Y, I)$ and so contributes $+1$ to $\overline{\tau}(Y, I) - \tau(X, I)$.

We have just proved that the density $\kappa(X, Y, t_1)$ of the road segment between X and Y at t_1 is correctly evaluated by Equation (39). With this the proof of Theorem Equation 6.0.5 is complete. □

6.0.3 MO3 SIMULATION USING NGSIM

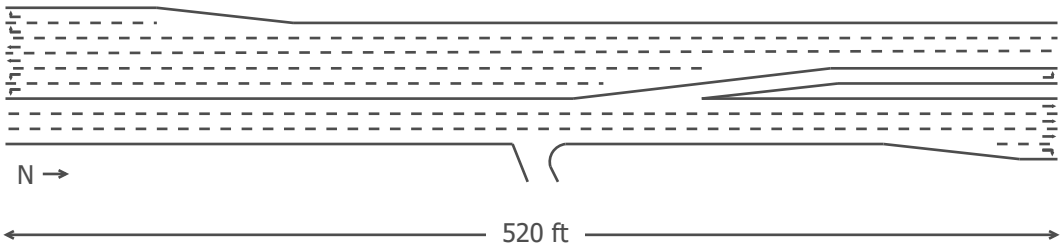


Fig. 46: Lankershim roadway

We start by testing the method using actual roadway traces provided by the NGSIM Lankershim dataset [124]. This dataset provides actual vehicle traces for Lankershim Boulevard, an urban arterial in Los Angeles, California. Vehicle trace data is provided for both northbound and southbound traffic every 100 milliseconds for thirty minutes. The stretch of road used in this simulation is the 520 feet between Universal Hollywood Drive to the south and Main Street to the north. There are three northbound lanes that split into a left turn lane, three straight lanes, and one right turn lane at the end of the test area. There are three southbound lanes that split into two left turn lanes, two straight lanes, and two right turn lanes at the end of the test area. The northbound lanes do include an exit within this stretch of road; however, it is used minimally by vehicles in the simulation. See Figure 46 for a visual representation of the simulation roadway.

When a vehicle enters into the roadway, it will begin recording the time of each passing vehicle. Then, upon passing another enabled vehicle in the oncoming traffic, the two will negotiate the start time as the time of latest vehicle entered the roadway. Using this start time, the two will determine their tallies and exchange them. The tallies will then be used, as well at their positions at the start time, to estimate the density using Equation (39).

The two vehicles will continue to work together. When the first one leaves the test area, they will again exchange tallies starting at the point they both passed one another. Then using the tallies and their positions at this time, they estimate the density again using Equation (39). The densities are saved and aggregated.

6.0.4 EVALUATION DETAILS

To verify our method using the NGSIM vehicle traces, we ran it ten times each for five

different penetration rates: 10%, 20%, 30%, 40%, and 50%. For each run, we average the results every one minute and use this value to estimate the density.

For each run, the number of vehicles passing the northbound entrance to the test area for each time period is counted and divided by the time period to get the actual density, labeled *Density Actual* in the plots. This actual density is used to compare with the measured density to analyze the accuracy of the method.

For each of the ten runs, we plot the minimum, 25% quartile, 75% quartile, and the maximum values of each of the ten runs using a period time of one minute. Additionally, the average of each of the ten runs is plotted as *Density Measured*. These plots are provided for 10%, 20%, 30%, 40%. and 50% in Figures 47(a), 47(b), 47(c), 47(d), and 47(e), respectively.

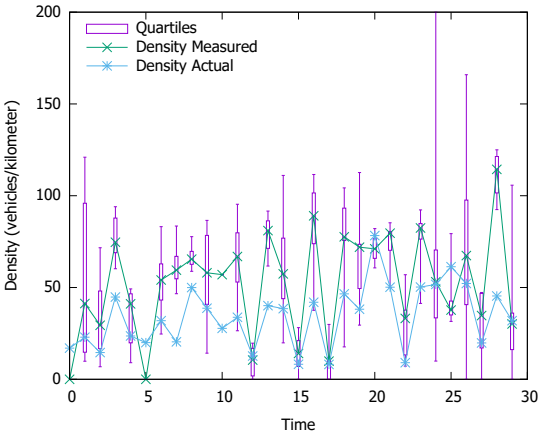
Finally Figure 47(f) provides a full view of the results. For each of the one minute increments, a Mean, 75% quartile, and maximum value is obtained. These are the Mean Relative Difference, the 75% Relative Difference, and the Max Relative Difference. Then the average values for each metric for each of the one minute increments is provided as the *Average Mean Relative Difference*, the *Average 75% Relative Difference*, and the *Average Max Relative Difference* which is plotted for each penetration rate. Finally, to show the worst values obtained, the maximum value of each of the Max Relative Differences is plotted as the *Max Max Relative Difference*.

6.0.5 SIMULATION RESULTS

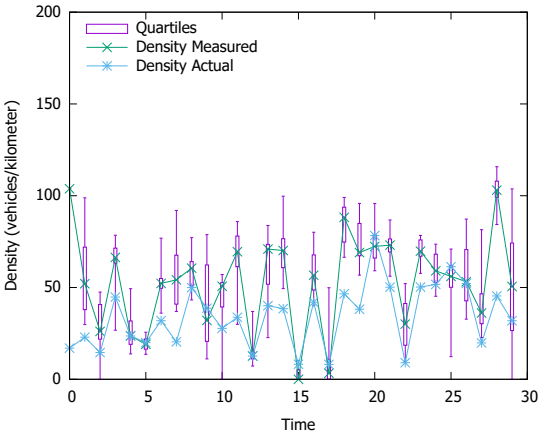
The *Average Mean Relative Difference* decreases from 97.3% with 10% penetration to 74.7% at 50% penetration. The *Average 75% Relative Difference* also decreases from 123.6% with 10% penetration to 101.7% with a 50% penetration. The *Average Max Relative Difference* decreases similarly, starting with 191.5% with 10% penetration to 133.6% with 50% penetration. Finally, the *Max Max Relative Difference*, which shows the maximum difference for each run starts with a relative difference of 530.6% with a 10% penetration down to 432.6% with a 50% penetration.

From the figures, as the penetration rates increase, it is clear the results for measured density converge towards a certain density value; however, this is not the actual density as reported. In fact, as reported in the previous paragraph, the difference between the measured and actual density values are very far off. In the next subsection we discuss why this is.

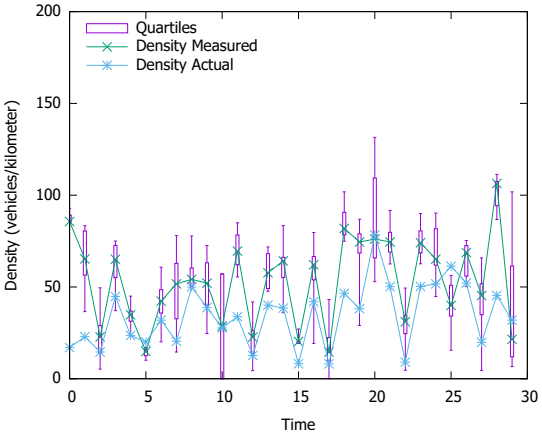
6.0.6 MO3 METHOD DISCUSSION



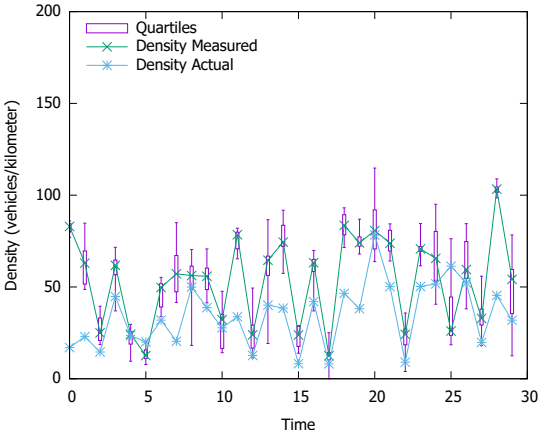
(a) NGSIM Lankershim 10% penetration



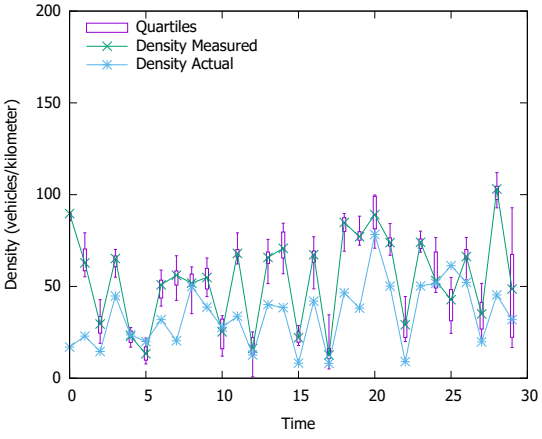
(b) NGSIM Lankershim 20% penetration



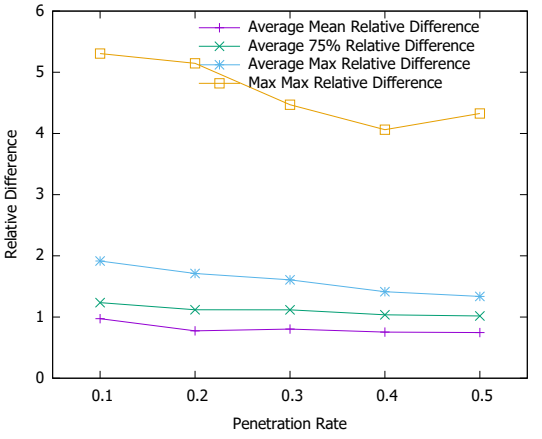
(c) NGSIM Lankershim 30% penetration



(d) NGSIM Lankershim 40% penetration



(e) NGSIM Lankershim 50% penetration



(f) NGSIM Lankershim Result Summary

Fig. 47: Results for NGSIM Lankershim traffic.

After carefully studying the data produced by the simulation runs, there are two issues that are causing the results to be so poor. First, an assumption was made that the northbound vehicle, X, and southbound vehicle, Y, will enter the intersection at the same time. Remember that both $\tau(X, I)$ and $\overline{\tau}(Y, I)$ share the same interval I. In reality and as shown by the simulation, it is rare for two vehicles to enter a roadway at exactly the same time. To handle this in the method, we allowed the two vehicles to negotiate the start time of the interval to be the later of the times the two vehicles entered the roadway.

To exemplify why this is an issue, suppose the northbound vehicle X enters the roadway from the south prior to Y. Additionally, suppose there are vehicles queued at the southbound traffic light. Vehicle X will pass by these vehicles and will not yet start counting them. Then, vehicle Y enters the test area, after vehicle X has passed those that have been queued. Vehicle X and Y will get a density measurement, but it will not include the entire intersection; even worse, it will likely exclude the vehicles that are at the ends of the roadway.

The second assumption is the main cause for the drastic differences in the measured and actual density. Remember that the NGSIM Lankershim is a dataset with urban traffic, which includes traffic lights. The traffic lights cause the variance in the density of traffic within a timespan to be quite large. Also, remember that an enabled vehicle will form a partnership with any enabled vehicles it meets. This means in times where the traffic is more dense, there is expected to be more enabled vehicles; therefore, it is expected that there will be more density measurements. Conversely, in times of less dense traffic there is expected to be less enabled vehicles; therefore, less density measurements.

When plotting the results, the density measurements are averaged for each minute. Since there are more density measurements when the density it high, this skews the average to be greater than the actual density.

We continue our validation of the MO3 method, but we have to consider the assumptions that are made. First, we want to consider traffic that is not queuing at the ends of the roadway. Additionally, we want to consider traffic where there are not wide differences in the density for each time period aggregated.

6.0.7 MO3 SIMULATION USING HIGHD

Our second dataset chosen is the HighD Dataset [126]. This dataset it not expected to have the same issues as the NGSIM Lankershim dataset from the previous section. The HighD Dataset is a short highway segment that does not include traffic signals, and there are not wide variances in the density. This dataset provides actual vehicle traces for German

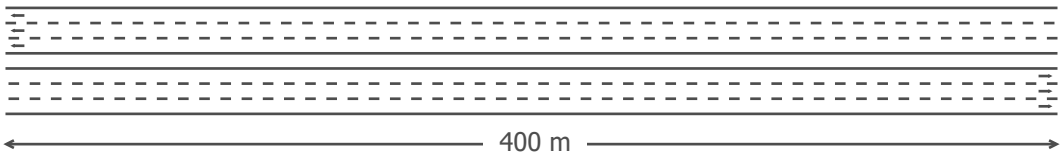


Fig. 48: HighD roadway

Highways by tracking vehicles using a drone flying over the roadway. Vehicle trace data is provided for traffic in both directions 25 times a second for 17 minutes. The roadway has three lanes in each direction and stretches 400 meters. See Figure 48 for a visual representation of the simulation roadway.

As with the NGSIM simulation, when a vehicle enters into the roadway, it will begin recording the time of each passing vehicle. Then, upon passing another enabled vehicle in the oncoming traffic, the two will negotiate the start time as the time of latest vehicle entered the roadway. Using this start time, the two will determine their tallies and exchange them. The tallies will then be used, as well at their positions at the start time, to estimate the density using Equation (39).

The two vehicles will continue to work together. When the first one leaves the test area, they will again exchange tallies starting at the point they both passed one another. Then using the tallies and their positions at this time, they estimate the density again using Equation (39). The densities are saved and aggregated.

6.0.8 EVALUATION DETAILS

To verify our method using the HighD vehicle traces, we ran it ten times each for five different penetration rates: 10%, 20%, 30%, 40%, and 50%. For each run, we average the results every one minute and use this value to estimate the density.

For each run, the number of vehicles passing the northbound entrance to the test area for each time period is counted and divided by the time period to get the actual density, labeled *Density Actual* in the plots. This actual density is used to compare with the measured density to analyze the accuracy of the method.

For each of the ten runs, we plot the minimum, 25% quartile, 75% quartile, and the

maximum values of each of the ten runs using a period time of one minute. Additionally, the average of each of the ten runs is plotted as *Density Measured*. These plots are provided for 10%, 20%, 30%, 40%. and 50% in Figures 49(a), 49(b), 49(c), 49(d), and 49(e), respectively.

Finally, Figure 49(f) provides a full view of the results. For each of the one minute increments, a Mean, 75% quartile, and maximum value is obtained. These are the Mean Relative Difference, the 75% Relative Difference, and the Max Relative Difference. Then the average values for each metric for each of the one minute increments is provided as the *Average Mean Relative Difference*, the *Average 75% Relative Difference*, and the *Average Max Relative Difference* which is plotted for each penetration rate. Finally, to show the worst values obtained, the maximum value of each of the Max Relative Differences is plotted as the *Max Max Relative Difference*.

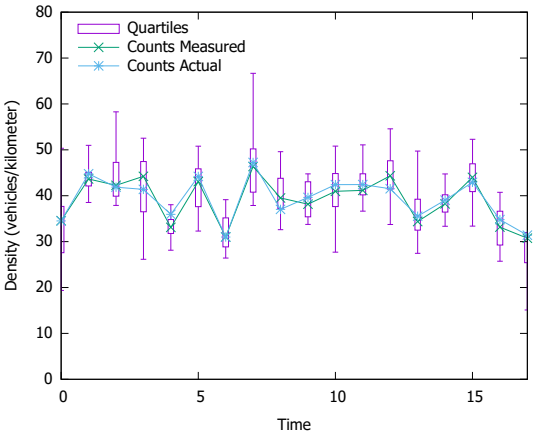
6.0.9 SIMULATION RESULTS

The *Average Mean Relative Difference* decreases from 11.2% with 10% penetration to 4.9% at 50% penetration. The *Average 75% Relative Difference* also decreases from 16.5% with 10% penetration to 6.5% with a 50% penetration. The *Average Max Relative Difference* decreases similarly, starting with 30.1% with 10% penetration to 10.1% with 50% penetration. Finally, the *Max Max Relative Difference*, which shows the maximum difference for each run starts with a relative difference of 51.9% with a 10% penetration down to 13.2% with a 50% penetration.

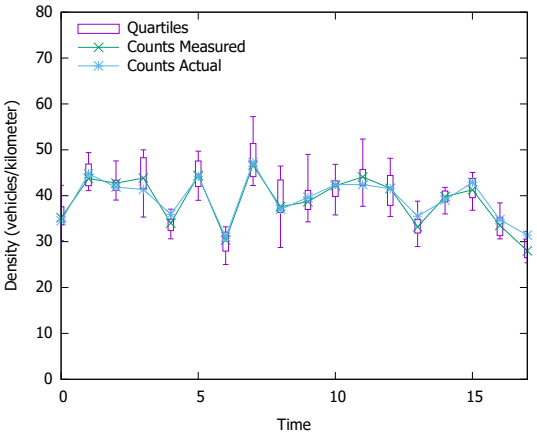
It is clear as the penetration rate increases, the variance in each of the run values converges to the actual density.

6.0.10 THE MO3 METHOD IN PRACTICE

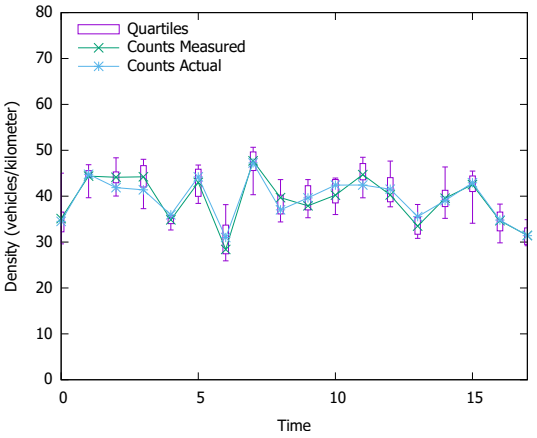
In the MO3 method there is a mix of enabled and non-enabled vehicles. The method uses enabled vehicles to account for non-enabled vehicles in both codirectional and oncoming traffic. Additionally, an enabled vehicle must have the ability to communicate wirelessly with other enabled vehicles and establish an agreement to run the MO3 method. The method is designed to work on short urban road segments, for example between two intersections; however, we also show the method will work in short stretches of highway as well. In both scenarios, there should be no occlusion between the two directions of traffic. Additionally, there should be no additional entrances or exits because of the conservation of flow requirement.



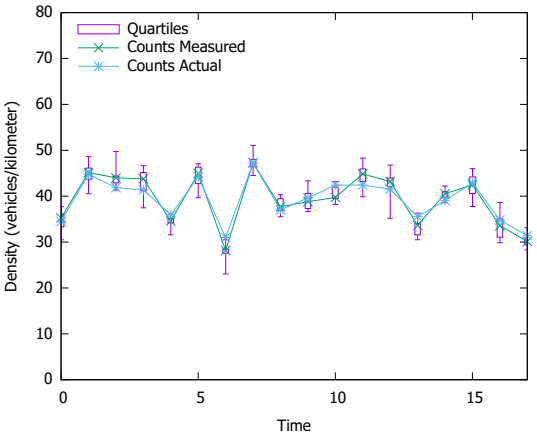
(a) HighD 10% penetration over two minutes



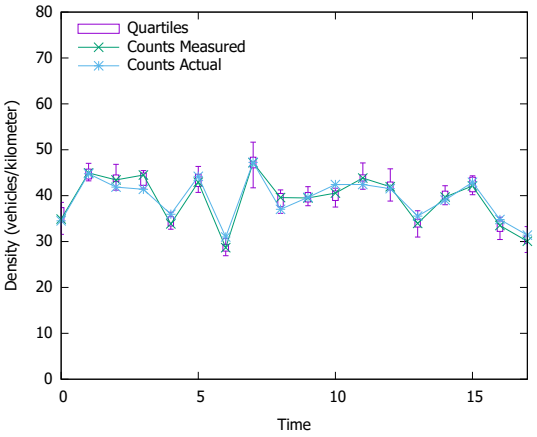
(b) HighD 20% penetration over two minutes



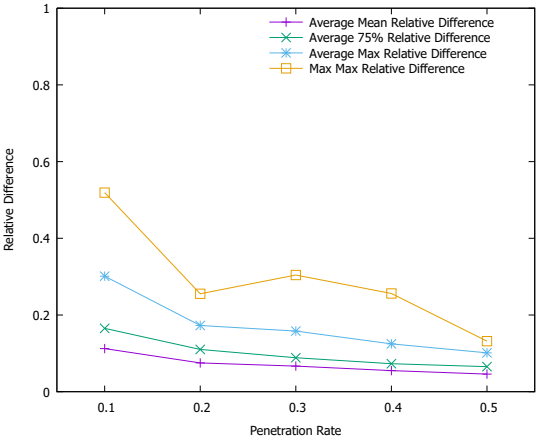
(c) HighD 30% penetration over two minutes



(d) HighD 40% penetration over two minutes



(e) HighD 50% penetration over two minutes



(f) HighD Result Summary

Fig. 49: Results for HighD traffic.

We now describe how the MO3 method will work using the messages and formats described in Section 3.3.

A TMC or local authority will establish a list of roadways that it requires density approximations using the MO3 method. It will assign a Test Area ID to each and establish the geometry of the roadway. It will then update RSUs or local agents near the test areas with the list of test areas.

The RSU or local agent will then regularly broadcast a WSA message using the established PSID for the test area advertisement service. The WSA will include details of the channel and details to receive WSM messages associated with test area advertisements.

The RSU or local agent will regularly broadcast the locations of nearby or upcoming test areas so vehicles within communication range are aware of the locations. This is done through the use of the Test Area Advertisement (TAA) messages described in Section 3.3.1. The TAA message will be sent using the WSM format and will contain the Test Area Id, the MO Method Type for the test area, and the geometry describing the boundaries of the test area.

An enabled vehicle will read the messages and locate the test area geometry on its digital map. Then, using its GPS and digital map, the vehicle will determine the moment it enters into a test area.

Upon entering a test area, the vehicle will advertise itself as an enabled vehicle. It will start the MO Partner Negotiation by sending a MO Partner Request (MPR) message, as described in Section 3.3.3. This message is a broadcast message sent using UPD over IPV6. This message will include the Test Area Id, the MO Method Type, and its own Temporary Id. Note that the Temporary ID used in the MPR message is the same that is used in the BSM message; this ensures any vehicles receiving the MPR can determine the location of the sender. The MPR message is sent regularly throughout the test area to ensure it can establish a MO Partner with any possible enabled vehicles.

When an enabled vehicle receives an MPR, it will check if it is within the same Test Area that is sent in the MPR message. If it is, it will then determine the location, velocity, and heading of the sender from a BSM message with the same Temporary ID as the MPR message. If both are traveling in the opposite direction toward one another, it will respond to the MPR with a MO Partner Accept (MPA) message. This message is described in Section 3.3.4. The vehicle receiving the MPR message knows the IPV6 address of the sender from the From Address of the UDP message. When responding it will send the MPR message using TCP over IPV6 and will send it to this same From Address. This message will include

the Test Area Id, Start Time, MO Method Type, and its own Temporary Id. The receipt and acknowledgment of the MPR message establishes that they are both part of an MO Partnership which starts at the Start Time. This Start Time will be the later time between the Start Time of the MPR message and the time the vehicle entered the test area. The Start Time from the MPA message will be the time used in the MO3 Method.

During the MO Partner Negotiation, each vehicle sends to the other its Temporary Id. Given that BSM messages are send regularly, each MO Partner will track the location, velocity, and heading of the other partner. Specifically, each MO Partner will store the IPv6 address, Temporary Id, and Start Time. Each time it receives a BSM from the Temporary ID of its partner it will update the location and the distance between the two of them.

Finally, it will maintain tallies of vehicles; however, with MO3, separate tallies are maintained for both codirectional and oncoming traffic. For codirectional traffic, the tally is the number of times a vehicle passes it minus the number of times it passes another vehicle. For oncoming traffic, the tally will be a count of the number of vehicles it meets.

Each partner will track the other using the BSM messages. Each will regularly estimate the time they will meet using the method discussed in Section 3.2.1. Note that this does not need to be done with every BSM message received; however, as the two get closer and closer, they will need to estimate the expected time they will pass more and more to establish the exact moment they meet.

Upon the moment they meet, each will create a Tally Exchange Message (TEM) as described in Section 3.3.5. The TEM will contain the data required to determine the density at the time the MO partnership was established using Equation (39) which is replicated below:

$$\kappa(X, Y, t_1) = \frac{\overline{\tau}(Y, I) - \tau(X, I)}{L}.$$

The required information to determine the density at the start time, t_1, is:

- $\overline{\tau}(Y, I)$, which is the tally from the oncoming vehicle,
- $\tau(X, I)$, which is the vehicle's own tally,
- L, which is the distance between the two vehicles at time t_1.

Sending the TEM represents the finish of the first phase of the MO3 method. The TEM includes a Tally Type field which represents that the density being measured is at the Start Time where the two MO partners were apart and traveling towards one another. The TEM

will include the following fields which are used by both vehicles to share the above data points so each can have an approximation for the density at the start time:

- Test Area ID - This is used to verify the test area being measured.
- Start Time - This is the start time as sent in the MPA message.
- End Time - This is the time that the vehicle determined as the time of passing. This value should be the same for both vehicles. If the two vehicles disagree on the End Time, there may be a need to reconcile the difference. This reconciliation is not in scope for this work.
- Start Location - This is the starting location of the vehicle.
- End Location - This is the final location of the vehicle. Both vehicles should have similar values; if they differ by a large amount there may be a need to reconcile the difference. Again, this reconciliation is not in scope for this work.
- Heading - This represents the heading of the vehicle.
- Tally Type - The tally type will be a 0 to represent that this is the first phase of the MO3 tally method. This is used to specify the density is measured at the Start Time and the Start Location.
- Tally Codirectional - This is the codirectional tally of the vehicle.
- Tally Oncoming - This is the oncoming tally of the vehicle.
- Flow - This field is left blank for the MO3 method.
- Density - This field is left blank for the MO3 method.

Each vehicle will send its TEM message using TCP over IPV6 with the respective IPV6 To and From address fields filled appropriately. After retrieving the message from its MO partner, each will determine the density between them at the start time using Equation (39).

Since each has an estimated density, both will create a Traffic Parameter Collection (TPC) message to send to the next RSU advertising the TPC service. The vehicle will listen for WSA messages that advertise the PSID associated to the TPC service. Once it receives one, it will create the TPC message and send it to the RSU via a WSM. The TPC message includes the following fields:

- Test Area ID - The Test Area ID identifies which Test Area the MO Measurement data is being reported.
- MO Method Type - The MO Method Type serves two purposes. First it is used to verify the method used, and second, it is used to determine the format of the MO Measurement Data.
- Start Location - The Start Location is the first field of the MO Measurement Data. It includes the location data of the vehicle at the time of the density measurement.
- End Location - The End Location field is the second field of the MO Measurement Data. It includes the location data of the vehicle's MO partner vehicle at the time of the density measurement.
- Time - The Time field is the third field of the MO Measurement Data. It is the time of the density measurement.
- Density - The Density field is the fourth and final field of the MO Measurement Data. It is the density between the Start Location and End Location at the time represented in the Time field.

This time represents a shift from the first phase of the MO3 method to the second phase. The second phase starts when the two enabled vehicles meet on the roadway and travel away from one another. It will end when the first of the two partner vehicles exits the test area. The old tallies from between the start time and the time they passed will be saved. New tallies starting at the time they both passed will be started. Each vehicle will continue to track the location of one another using BSM messages and will determine the distance between them. Upon reaching the end of the test area, the first vehicle will generate a TEM message using its tallies for the second measurement.

The TEM will include the following fields which are used by both vehicles to share the above data points so each can have an approximation for the density at the start time:

- Test Area ID - This is used to verify the test area being measured.
- Start Time - This is the time the two vehicles passed, thus starting the second measurement.
- End Time - This is the current time, when the vehicle exited the test area.

- Start Location - This is the location the two vehicles passed to start the second measurement.
- End Location - This is the current location of the vehicle.
- Heading - This represents the heading of the vehicle.
- Tally Type - The tally type will be a 1 to represent that this is the second phase of the MO3 tally method. This is used to specify the density is measured at the End Time and the End Location.
- Tally Codirectional - This is the codirectional tally of the vehicle.
- Tally Oncoming - This is the oncoming tally of the vehicle.
- Flow - This field is left blank for the MO3 method.
- Density - This field is left blank for the MO3 method.

The MO partner, upon receiving the second TEM will immediately generate its own TEM and reply with the following TEM message:

- Test Area ID - This is used to verify the test area being measured.
- Start Time - This is the time the two vehicles passed, thus starting the second measurement.
- End Time - This is the same start time from the partner's TEM message.
- Start Location - This is the location the two vehicles passed to start the second measurement.
- End Location - This is the location of the vehicle at the End Time.
- Heading - This represents the heading of the vehicle.
- Tally Type - The tally type will be a 1 to represent that this is the second phase of the MO3 tally method. This is used to specify the density is measured at the End Time and the End Location.
- Tally Codirectional - This is the codirectional tally of the vehicle.
- Tally Oncoming - This is the oncoming tally of the vehicle.

- Flow - This field is left blank for the MO3 method.
- Density - This field is left blank for the MO3 method.

Again, both will have an estimate for the density, both will create a Traffic Parameter Collection (TPC) message to send to the next RSU advertising the TPC service. The vehicle will listen for WSA messages that advertise the PSID associated to the TPC service. Once it receives one, it will create the TPC message and send it to the RSU via a WSM. The TPC message includes the following fields:

- Test Area ID - The Test Area ID identifies which Test Area the MO Measurement data is being reported.
- MO Method Type - The MO Method Type serves two purposes. First, it is used to verify the method used, and second, it is used to determine the format of the MO Measurement Data.
- Start Location - The Start Location is the first field of the MO Measurement Data. It includes the location data of the vehicle at the time of the density measurement.
- End Location - The End Location field is the second field of the MO Measurement Data. It includes the location data of the vehicle's MO partner vehicle at the time of the density measurement.
- Time - The Time field is the third field of the MO Measurement Data. It is the time of the density measurement.
- Density - The Density field is the fourth and final field of the MO Measurement Data. It is the density between the Start Location and End Location at the time represented in the Time field.

The MO3 method is complete when either vehicle exits the test area and the TEM messages are both exchanged.

6.0.11 DISCUSSION

As mentioned previously, the MO3 method has two main issues when being used to measure the density of traffic in an urban arterial that has a wide variance in traffic, such as that introduced by a traffic signal. We have explained the issues are being caused by two

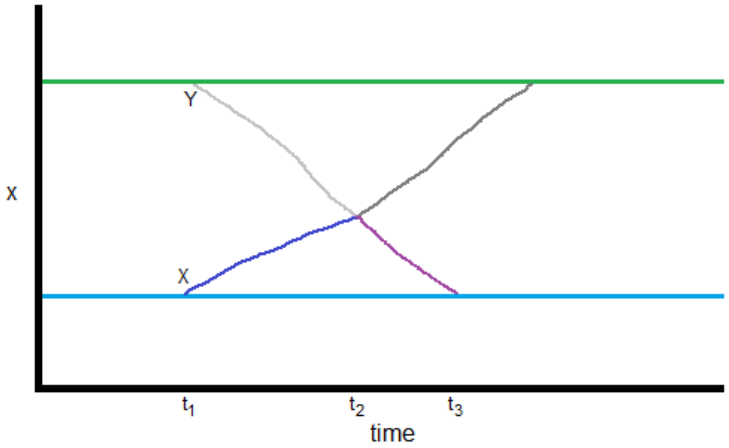
assumptions. The first assumption is that both partners in oncoming traffic are assumed to both enter the roadway at the same time. This is clearly not a realistic assumption. The second assumption is that averaging the traffic over a particular time span will match the actual density of the roadway. As explained previous, the MO3 method will have more density measurements in times of denser traffic and less density measurements in times of lighter traffic. When averaging the density measurements, the average is skewed to be too high.

The next method considers a different metric and a different method that mitigates these issues and allows the method to work in urban traffic.

6.1 MO3 - FLOW

6.1.1 INTRODUCTION

(a) Roadway view



(b) Time-space view

Fig. 50: Illustrating the MO3 flow method.

The MO3-Flow method was inspired by the failure of the MO3 method in urban roadways. As discussed in the previous Section 6.0.11, it is not common that two enabled vehicles enter a test area at exactly the same time. Additionally, the results need to be aggregated in a way that does not skew the results towards the results taken for denser traffic. This method shares a lot of similarities to MO3, except that it measures flow instead of density and that it uses a different means of aggregating the results.

Use Figure 50 as an illustration of the MO1 method. There are two figures representing two different views of the traffic. In Figure 50(a) shows a top down view of vehicles X and Y on the road and Figure 50(b) shows the time-space view of the vehicles. Consider a vehicle X traveling in the direction of interest. As it enters into the test area at t_1 it will maintain a tally of codirectional traffic. At time t_1 vehicle X meets oncoming vehicle Y, where it will send its tally to Y. Vehicle Y will then maintain a tally of oncoming traffic until it reaches the end of the test area at time t_3. Note that in the method there are two phases. In the first phase, the first vehicle keeps a tally. Then in the second phase the tally is passed to the second vehicle and it takes over. This is different from the MO3 method where both vehicles are keeping the tallies at the same time. The tallies are used to measure the number of vehicles passing into the test area between time t_1 and time t_3. With the number and the interval, the flow of vehicles can be determined.

6.1.2 PRELIMINARIES

We assume a multi-lane urban arterial road. In this setting, we focus our attention on evaluating the local traffic flow between two consecutive traffic lights. The arterial is populated by vehicles that travel in both directions.

Consider a stretch of road between two points p and q. Additionally, consider two time intervals $I = [t_1, t_2]$ with $t_1 < t_2$ and $J = [t_2, t_3]$ with $t_2 < t_3$ and vehicles X and Y moving in opposite directions. t_1 is the time at which X passes point p, t_2 is the time at which X meets Y, and t_3 is the time at which Y passes point p. Note that X must meet Y between points p and q. We assume that at time t_1, $x(t_1) < y(t_1)$ and at time t_3, $x(t_3) > y(t_3)$

We find it useful to categorize the vehicles *codirectional* with X in terms of their relative position with respect to X and Y in the time interval I and in time interval J.

We now take another look at Definitions 4, 5, 6, and 7; however, we restate Definition 5 with a more restrictive definition for this method; notice that it is now listed as Definition 11.

Definition 4 *We say that vehicle C, codirectional with X, is of Type 1 with respect to X*

in I if C is behind X at t_1 and X is behind C at t_2.

Refer to Figure 51 for an illustration. Observe that Definition 4 requires C to pass X at least once in I; as observed above, C may pass X at any time in I, except at t_2.

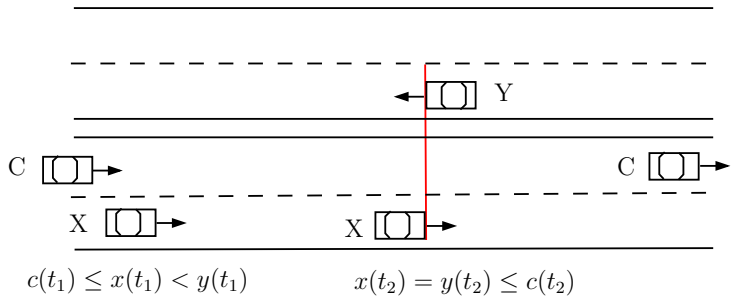


Fig. 51: C is of Type 1 with respect to X in I.

Definition 11. *We say that vehicle C, codirectional with X, is of Type 2 with respect to X in I if C is behind X at both t_1 and t_2 and vehicle C has passed by p prior to t_3.*

Refer to Figure 52 for an illustration. Observe that C may or may not pass X in I.

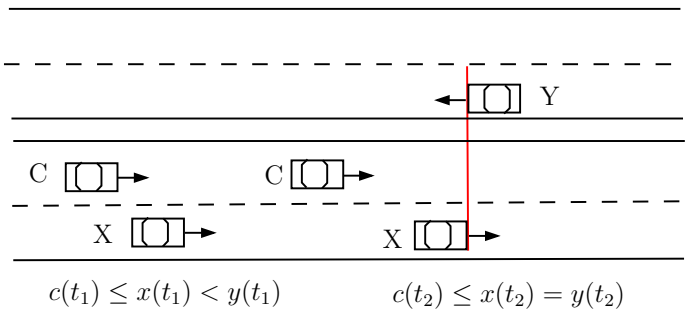


Fig. 52: C is of Type 2 with respect to X in I.

Definition 6 *We say that vehicle C, codirectional with X, is of Type 3 with respect to X in I if X is behind C at t_1, $c(t_1) \leq y(t_1)$ and X is behind C at t_2.*

Refer to Figure 53 for an illustration. Observe that X may or may not pass C in I.

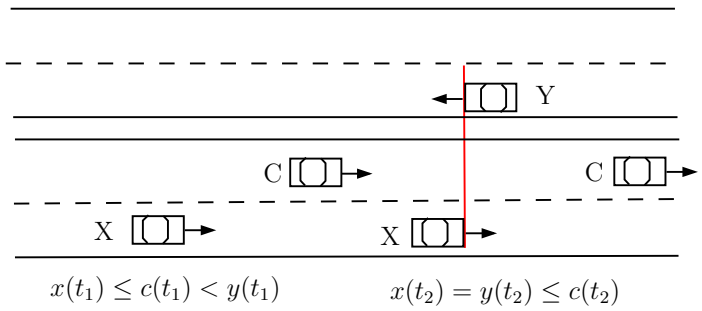


Fig. 53: C is of Type 3 with respect to X in I.

Definition 7 *We say that vehicle C, codirectional with X, is of Type 4 with respect to X in I if X is behind C at t_1 and C is behind X at t_2.*

Refer to Figure 54 for an illustration. Observe that Definition 7 implies $c(t_1) < y(t_1)$ and that X must pass C at least once in I; as observed above, X may pass C at any interior point of I, but not at t_2.

Again, Definition 8 is restated as it is important for this method.

Definition 8 *Consider a generic vehicle X and let $n_f(X, I)$ denote the number of times X was passed by other codirectional vehicles in I. Similarly, let $n_s(X, I)$ stand for the number of times X passed other codirectional vehicles in I. We write*

$$\tau(X, I) = n_f(X, I) - n_s(X, I) \tag{40}$$

and refer to $\tau(X, I)$ as the tally *of X over the time interval I.*

For this method, we also redefine Definition 10 in the context of the method.

Definition 12. *Let $\overline{\tau}(Y, J)$ be the number of* oncoming *vehicles that have met Y in the time interval J. At the risk of mild confusion, we refer to $\overline{\tau}(Y, J)$ as the* tally *of Y in J.*

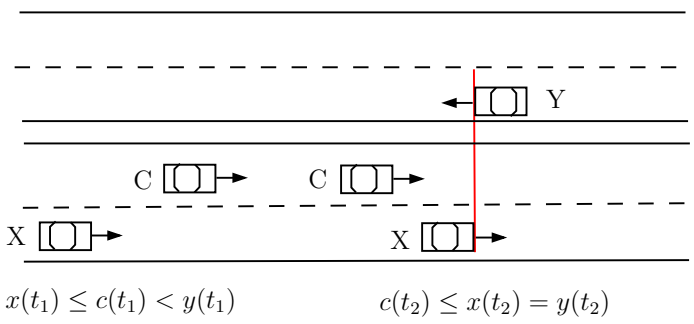


Fig. 54: C is of Type 4 with respect to X in I.

One of the main contributions of this work is to show that the flow can be computed in terms of the tallies $\tau(X, I)$ and $\overline{\tau}(Y, J)$ of X and Y. In order to state our main result, we need a number of intermediate results that will be developed in Section 6.1.3.

6.1.3 TECHNICALITIES

As it turns out, the tally concept in Definitions 8, from Section 4.2, and 12 is fundamental to approximating traffic flow. We begin by stating and proving a number of technical results.

Lemmas 4.2.3, 4.2.4, and 4.2.5 are restated given their importance to this method.

Lemma 4.2.3 *Every vehicle of Type 1 with respect to X in I contributes* $+1$ *to* $\tau(X, I)$.

Lemma 4.2.4 *Every vehicle of Type 2 or Type 3 with respect to X in I contributes* 0 *to* $\tau(X, I)$.

Lemma 4.2.5 *Every vehicle of Type 4 with respect to X in I contributes* -1 *to* $\tau(X, I)$.

We now restate the results for oncoming traffic.

Lemma 6.1.1. *Every vehicle of Type 1 with respect to X in I contributes* 0 *to* $\overline{\tau}(Y, J)$.

Proof. Consider an arbitrary vehicle C of Type 1 with respect to X in I. By definition, X is behind C at t_2 and, consequently, since X and Y meet at time t_2, C will already have been met by Y prior to the start of interval J. It follows that C contributes 0 to $\overline{\tau}(Y, J)$. □

Lemma 6.1.2. *Every vehicle of Type 2 with respect to X in I contributes* $+1$ *to* $\overline{\tau}(Y, J)$.

Proof. Consider an arbitrary vehicle C of Type 2 with respect to X in I. By definition, C is behind X at time t_2 and has passed p_1 prior to t_3. As a result, C will meet Y in J. It

follows that C contributes $+1$ to $\overline{\tau}(Y, J)$. □

Lemma 6.1.3. *Every vehicle of Type 3 with respect to X in I contributes* 0 *to* $\overline{\tau}(Y, J)$.

Proof. Consider an arbitrary vehicle C of Type 3. By definition, X is behind C at t_2 and, consequently, since X and Y meet at time t_2, C will already have been met by Y prior to the start of interval J. It follows that C contributes 0 to $\overline{\tau}(Y, J)$. □

Lemma 6.1.4. *Every vehicle of Type 4 with respect to X in I contributes* $+1$ *to* $\overline{\tau}(Y, J)$.

Proof. Consider an arbitrary vehicle C of Type 4 with respect to X in I. By definition, C has passed point p_1 prior to X and C is behind X at time t_2 and, as a result, C will meet Y in J. It follows that C contributes $+1$ to $\overline{\tau}(Y, J)$. □

Lemma 6.1.5. *Every vehicle of Type 1 with respect to X in I will pass p in I.*

Proof. Consider an arbitrary vehicle C of Type 1 with respect to X in I. By definition, C is behind X at t_1, the time at which X passes point p. Since C must pass X in I to be Type 1, it must also pass point p in I. The proof is complete. □

Lemma 6.1.6. *Every vehicle of Type 2 with respect to X in I will pass p in I or J.*

Proof. By definition each vehicle of Type 2 must be behind X at t_1. This means it cannot pass point p until after t_1. Additionally, by definition each vehicle of Type 2 must pass p in prior to t_3, the time at which interval J ends. It follows that each Type 2 vehicle must pass p in I or J. □

Lemma 6.1.7. *Any vehicle of Type 3 with respect to X in I will not pass p in I or J.*

Proof. Consider an arbitrary vehicle C of Type 3 with respect to X in I. By definition, C is front of X at t_1, the time at which X passes point p. Since C has already passed p prior to t_1, it will not pass p in I or J. The proof is complete. □

Lemma 6.1.8. *Any vehicle of Type 4 with respect to X in I will not pass p in I or J.*

Proof. Consider an arbitrary vehicle C of Type 4 with respect to X in I. By definition, C is front of X at t_1, the time at which X passes point p. Since C has already passed p prior to t_1, it will not pass p in I or J. The proof is complete. □

We are now in a position to state and prove the main results of this paper. Specifically, we are interested in the number of vehicles passing point p in some interval I. We denote this as $N(p, I)$.

Theorem 6.1.9. *Consider a time interval $I = [t_1, t_2]$ with $t_1 < t_2$, a time interval $J = [t_2, t_3]$ with $t_2 < t_3$, and vehicles X and Y traveling in opposite direction where X passes point p at time t_1, X meets Y at time t_2, and Y passes p at time t_3. The number of vehicles passing point p between time t_1 and t_3 is*

$$N(p, [t_1, t_3]) = \tau(X, I) + \overline{\tau}(Y, J). \tag{41}$$

Proof. We are interested in the number of vehicles passing point p between time t_1 and t_3. We prove this by considering, case by case, each of the types of vehicles.

First, consider an arbitrary vehicle of Type 1. By Lemmas 4.2.3 and 6.1.1 this vehicle contributes $+1$ to $\tau(X, I)$ and 0 to $\overline{\tau}(Y, J)$, thus contributing $+1$ to $\tau(X, I) + \overline{\tau}(Y, J)$. Also, by Lemma 6.1.5, each Type 1 vehicle will pass p in I contributing $+1$ to $N(p, [t_1, t_3])$.

Next, consider an arbitrary vehicle of Type 2. By Lemmas 4.2.4 and 6.1.2 this vehicle contributes 0 to $\tau(X, I)$ and $+1$ to $\overline{\tau}(Y, J)$, thus contributing $+1$ to $\tau(X, I) + \overline{\tau}(Y, J)$. Additionally, by Lemma 6.1.6, each Type 2 vehicle will pass p in I or J contributing $+1$ to $N(p, [t_1, t_3])$.

Further, consider an arbitrary vehicle of Type 3. By Lemmas 4.2.4 and 6.1.3 this vehicle contributes 0 to $\tau(X, I)$ and 0 to $\overline{\tau}(Y, J)$, thus contributing 0 to $\tau(X, I) + \overline{\tau}(Y, J)$. By Lemma 6.1.7, each Type 3 vehicle will not pass p in I or J contributing 0 to $N(p, [t_1, t_3])$.

Finally, consider an arbitrary vehicle of Type 4. By Lemmas 4.2.5 and 6.1.4 this vehicle contributes -1 to $\tau(X, I)$ and $+1$ to $\overline{\tau}(Y, J)$, thus contributing 0 to $\tau(X, I) + \overline{\tau}(Y, J)$. By Lemma 6.1.8, each Type 4 vehicle will not pass p in I or J contributing 0 to $N(p, [t_1, t_3])$.

In each case, the vehicle types contribution to $\tau(X, I) + \overline{\tau}(Y, J)$ matches the vehicle types contribution to $N(p, [t_1, t_3])$, proving Equation (41). With this the proof of Theorem 6.1.9 is complete. □

With this result, we know the number of vehicles passing through point p between time t_1 and t_3. We can extend this result by calculating the flow of vehicles through p in the time interval $[t_1, t_3]$ by simply dividing $N(p, [t_1, t_3])$ over the length of the time interval.

$$q(p, [t_1, t_3]) = \frac{N(p, [t_1, t_3])}{t_3 - t_1}. \tag{42}$$

6.1.4 USING THE METHOD

In this section we start by showing how the result of Equation (42) can be used in practice and then in Section 6.1.5 we show how to aggregate results to get a more granular

measurement of the flow. Our method will work with vehicles traveling in any direction; however, for illustration and to reduce confusion, we will use the terms northbound and southbound vehicles to describe the direction of vehicles. Additionally, the method will work in either direction, but to simplify we will only describe measuring northbound traffic.

Let us consider an enabled northbound vehicle X that crosses point p at time t_0. X will keep a tally of all northbound vehicles, as defined in Definition 8. Upon meeting an enabled southbound vehicle Y at some t_i, X will have a complete tally, $\tau(X, [t_0, t_i])$. X will transmit this tally to Y, and Y will start its own tally of northbound vehicles, as defined in Definition 12, until it passes p at time t_1. Upon passing p, Y will have its complete tally, $\overline{\tau}(Y, [t_i, t_1])$. Using Equation (42), Y will use both tallies to compute $q(p, [t_0, t_1])$, the flow at point p between time interval $[t_0, t_1]$.

6.1.5 AGGREGATING THE RESULTS

Given a set of values of N each with the same p, we are interested in aggregating the results. We refer to this set as M.

Remember, given Theorem 6.1.9, the number of vehicles passing point p in the time interval $[t_1, t_2]$ is $N(p, [t_1, t_2])$. Note that the start of the interval is represented by a northbound vehicle and the end of the interval is represented by a southbound vehicle.

Additionally, we must introduce some terminology prior to the main theorem needed for aggregation. Two vehicles are said to be *sequential* if there are no other enabled vehicles passing point p between them.

Consider any two values of N, N_1 and N_2, defined by $N_1(p, [t_{i1}, t_{j1}])$ and $N_2(p, [t_{i2}, t_{j2}])$ where $t_{i1} < t_{i2}$. N_1 and N_2 are said to be *connected* if $t_{i1} <= t_{i2} <= t_{j1}$. We overload the term by also referring to the interval $[t_{i1}, t_{j2}]$ as being *connected*. Additionally, within a set M, if there exists any interval that is not connected, we refer to this interval as a *gap*.

We now move on to the main theorem we will use for aggregation.

Theorem 6.1.10. *Given a connected set of values of N at point p, the number of vehicles that pass by point p can be determined between any two enabled vehicles that pass by point p sequentially.*

Proof. We prove this by considering the following four cases:

- Case 1. The count between sequential northbound vehicles,
- Case 2. The count between a sequential north and southbound vehicle,

- Case 3. The count between a sequential south and northbound vehicle, and
- Case 4. The count between sequential southbound vehicles.

Case 1. The count between sequential northbound vehicles

Suppose n northbound vehicles, X_1, X_2, ..., X_n, that pass point p sequentially at time t_1, t_2, ..., t_n, respectively. Each meet the same southbound vehicle Z which travels southbound and passes point p at time t_z. The value N for each pair is denoted as $N(p, [t_1, t_z])$, $N(p, [t_2, t_z])$, ..., $N(p, [t_n, t_z])$.

The number of northbound vehicles that pass point p between any two northbound sequential vehicles X_i and X_{i+1} is denoted as $N(p, [t_i, t_{i+1}]) = N(p, [t_{i+1}, t_z]) - N(p, [t_i, t_z])$, where $1 \leq i < n$. This proves the count between sequential northbound vehicles can be determined.

Case 2. The count between a sequential north and southbound vehicle

Suppose a northbound vehicle, X, passes point p at time t_1 and meets a southbound vehicle Z. Z travels southbound and passes point p at time t_2. The value N for each pair is denoted as $N(p, [t_1, t_2])$. This case is already proved in Theorem 6.1.9.

Case 3. The count between a sequential south and northbound vehicle

Suppose a southbound vehicle Z passes point p at time t_1, and then a northbound vehicle X passes point p at time t_2. t_1 represents the end time of a value of N, denoted as N_1, and t_2 represents the time of a another value of N, denoted as N_2. In order to be connected, there must exist another value of N, N_i, such that N_i overlaps both N_1 and N_2; otherwise, N_1 and N_2 would not be connected or Z and X would not be sequential.

N_i is determined by a northbound vehicle X_i that passes point p at time t_s, where $t_s \leq t_1$. X_i meets vehicle Z_i and transmits its tally. Then Z_i passes point p at a time t_f, where $t_f \geq t_2$. As such N_i can be fully denoted as $N_i(p, [t_s, t_f])$.

Notice that X_i must have passed Z at some point in $[t_1, t_s]$ so there must exist a value $N(p, [t_1, t_s])$. Additionally, X must have passed Z_i at some point in $[t_2, t_f]$, so there must exist a value $N(p, [t_2, t_f])$.

The count of vehicles between the south and northbound vehicles is denoted as $N(p, [t_1, t_2]) = N(p, [t_s, t_t]) - N(p, [t_s, t_1]) - N(p, [t_2, t_f])$. The proof for this case is complete.

Case 4. The count between sequential southbound vehicles

Suppose a northbound vehicle X passes point p at time t_0. Then X meets n southbound vehicles Z_1, Z_2, ..., Z_n, that pass point p sequentially at time t_1, t_2, ..., t_n, respectively. The value N for each pair is denoted as $N(p, [t_0, t_1])$, $N(p, [t_0, t_2])$, ..., $N(p, [t_0, t_n])$.

The number of northbound vehicles that pass point p between any two sequential vehicles Z_i and Z_{i+1} is denoted as $N(p, [t_i, t_{i+1}]) = N(p, [t_0, t_{i+1}]) - N(p, [t_0, t_i])$, where $1 \leq i < n$. This proves the count between sequential southbound vehicles can be determined.

With each of the cases proved, we have shown that number of vehicles between any two sequential enabled vehicles given the associated connected set of N.

□

We note that with the above aggregations, the value $N(p, I)$ no longer represents the time between a northbound vehicle passing p and a southbound vehicle passing p. It will be the time between any sequential vehicle passing p. To represent this difference, we will use N to represent the original values and M to represent a set of N. We will use N' to represent an aggregated value of N and M' to represent a set of N'.

6.1.6 ESTIMATING FLOW OVER REGULAR INTERVALS

As defined in Equation (1), Flow is the number of vehicles that pass a point p on the roadway per unit of time. It is common in practice to report the flow for regular periods of time. This poses a problem for our method since it would be atypical to expect vehicles cross a certain point p according to an observers requirement.

In this section we are interested in the number of vehicles passing point p in the interval $[\tau_1, \tau_2]$. There will be some values of N' in M' that fully fit within the interval and there may be some that overlap with the interval. We need a means to estimate only those that pass point p within the interval $[\tau_1, \tau_2]$.

To do this we can assume a general distribution function F for vehicles passing point p.

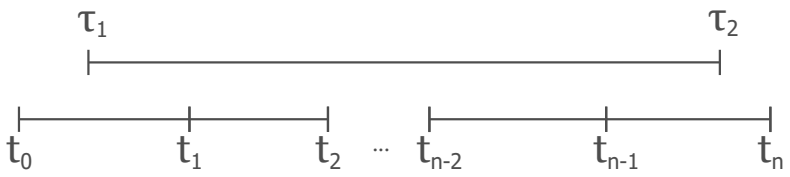


Fig. 55: Illustrating aggregation

Referring to Figure 55, suppose a set of M' containing the following: $N'(p, [t_0, t_1]), N'(p, [t_1, t_2]), ...,$

$N'(p,[t_{n-2},t_{n-1}])N'(p,[t_{n-1},t_n])$. Additionally, suppose τ_1 is between t_0 and t_1, and τ_2 is between t_{n-1} and t_n.

The vehicles in $N'(p,[t_0,t_1])$ that are in the interval $[\tau_1,t_1]$ is

$$N'(p,[\tau_1,t_1]) = N'(p,[t_0,t_1]) \cdot (F(t_1) - F(\tau)) \tag{43}$$

Similarly, the vehicles in $N'(p,[t_{n-1},t_n])$ that are in the interval $[t_{n-1},\tau_2]$ is

$$N'(p,[t_{n-1},\tau_2]) = N'(p,[t_{n-1},t_n]) \cdot F(\tau_2) \tag{44}$$

The value of N' that fit fully within the interval $[\tau_1,\tau_2]$ is:

$$N'(p,[t_1,t_{n-1}]) = N'(p,[t_1,t_2]) + ... + N'(p,[t_{n-2},t_{n-1}]) \tag{45}$$

Equations (43), (44), and (45) can be added together to get:

$$\begin{aligned} N'(p,[\tau_1,\tau_2]) = {} & N'(p,[t_0,t_1]) \cdot (F(t_1) - F(\tau)) \\ & + N'(p,[t_1,t_2]) + ... + N'(p,[t_{n-2},t_{n-1}]) \\ & + N'(p,[t_{n-1},t_n]) \cdot F(\tau_2) \end{aligned} \tag{46}$$

Now, if we assume the distribution function F is uniformly distributed, Equation (46) can write this as:

$$\begin{aligned} N'(p,[\tau_1,\tau_2]) = {} & N'(p,[t_0,t_1]) \frac{t_1 - \tau_1}{t_1 - t_0} \\ & + N'(p,[t_1,t_2]) + ... + N'(p,[t_{n-2},t_{n-1}]) \\ & + N'(p,[t_{n-1},t_n]) \frac{\tau_2 - t_{n-1}}{t_n - t_{n-1}} \end{aligned} \tag{47}$$

6.1.7 MO3-FLOW SIMULATION

The MO3-Flow method was tested using the same NGSIM Lanksershim dataset as described in Section 6.0.3.

In each of the experiments, when an enabled northbound vehicle moves into the test area from the south, it will start a tally. Then upon meeting an enabled southbound vehicle the northbound vehicle will send it its tally. When the southbound vehicle moves out of the test area to the south, it will send its value for N using Equation (41) and its time interval to a

road side unit. The road side unit will collect and aggregate values of N using the method discussed in Section 6.1.6.

6.1.8 EVALUATION DETAILS

To verify our method using the NGSIM vehicle traces, we ran it ten times each for five different penetration rates: 10%, 20%, 30%, 40%, and 50%. For each run, we average the results every two minutes and use this value to estimate the flow.

For each run, the number of vehicles passing the northbound entrance to the test area for each time period is counted and divided by the time period to get the actual flow, labeled *Flow Actual* in the plots. This actual flow is used to compare with the measured flow to analyze the accuracy of the method.

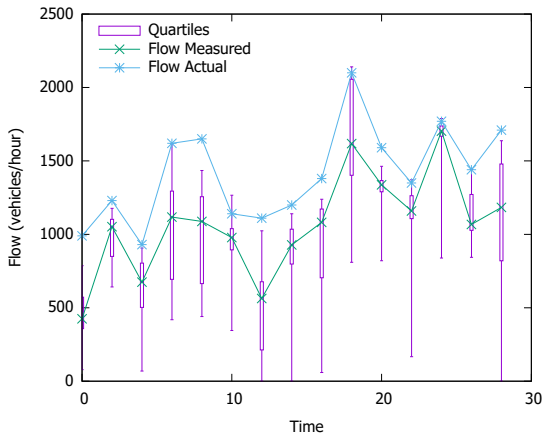
For each of the ten runs, we plot the minimum, 25% quartile, 75% quartile, and the maximum values of each of the ten runs using a period time of two minutes. Additionally, the average of each of the ten runs is plotted as *Flow Measured*. These plots are provided for 10%, 20%, 30%, 40%. and 50% in Figures 56(a), 56(b), 56(c), 56(d), and 56(e), respectively.

Finally Figure 56(f) provides a full view of the results. For each of the two minutes increments, a Mean, 75% quartile, and maximum value is obtained. These are the Mean Relative Difference, the 75% Relative Difference, and the Max Relative Difference. Then the average values for each metric for each of the two minute increments is provided as the *Average Mean Relative Difference*, the *Average 75% Relative Difference*, and the *Average Max Relative Difference* which is plotted for each penetration rate. Finally, to show the worst values obtained, the Maximum value of each of the Max Relative Differences is plotted as the *Max Max Relative Difference*.

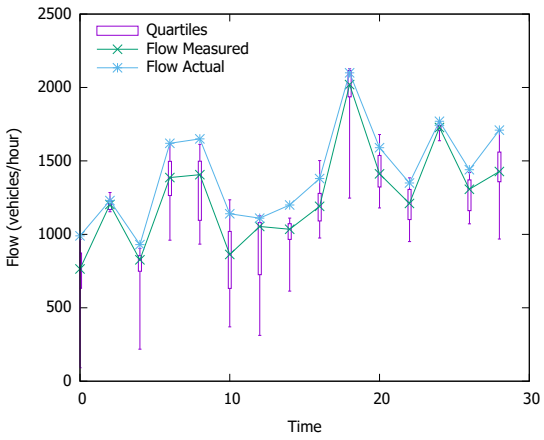
6.1.9 SIMULATION RESULTS

The *Average Mean Relative Difference* decreases from 23.3% with 10% penetration to 5.7% at 50% penetration. The *Average 75% Relative Difference* also decreases from 38.2% with 10% penetration to 8.6% with a 50% penetration. The *Average Max% Relative Difference* decreases similarly, starting with 74.6% with 10% penetration to 16.3% with 50% penetration. Finally, the *Max Max Relative Difference*, which shows the maximum difference for each run starts with a relative difference of 100% with a 10% penetration down to 35.3% with a 50% penetration.

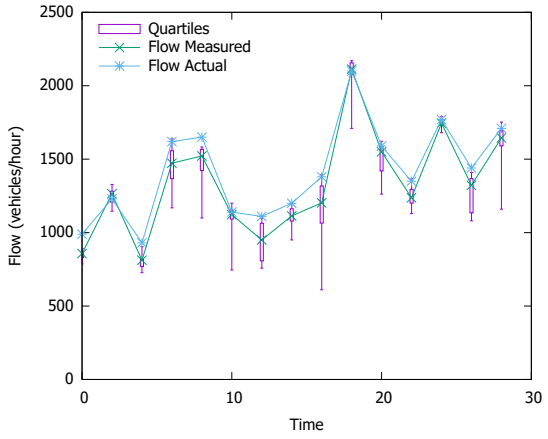
It is clear as the penetration rate increases, the variance in each of the run values converges on the actual flow. This is because the number of gaps in the measurements



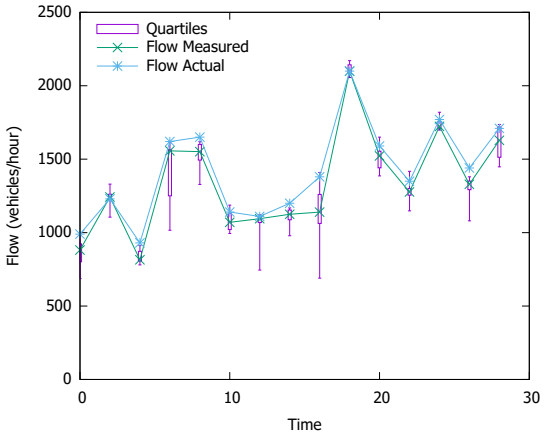
(a) NGSIM Lankersim 10% penetration over two minutes



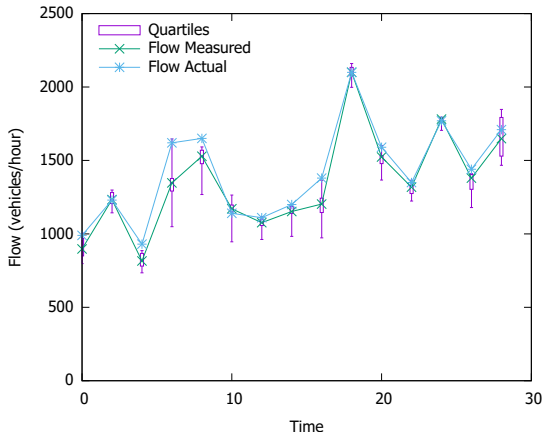
(b) NGSIM Lankersim 20% penetration over two minutes



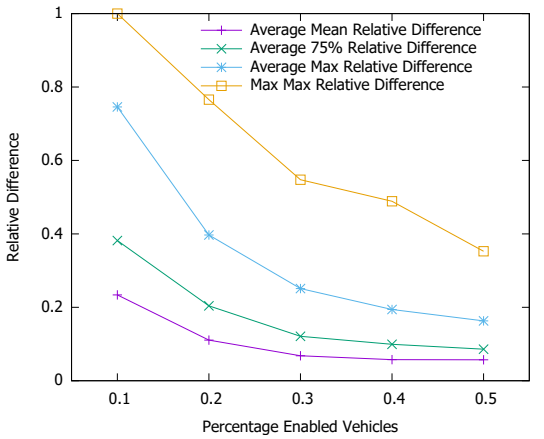
(c) NGSIM Lankersim 30% penetration over two minutes



(d) NGSIM Lankersim 40% penetration over two minutes

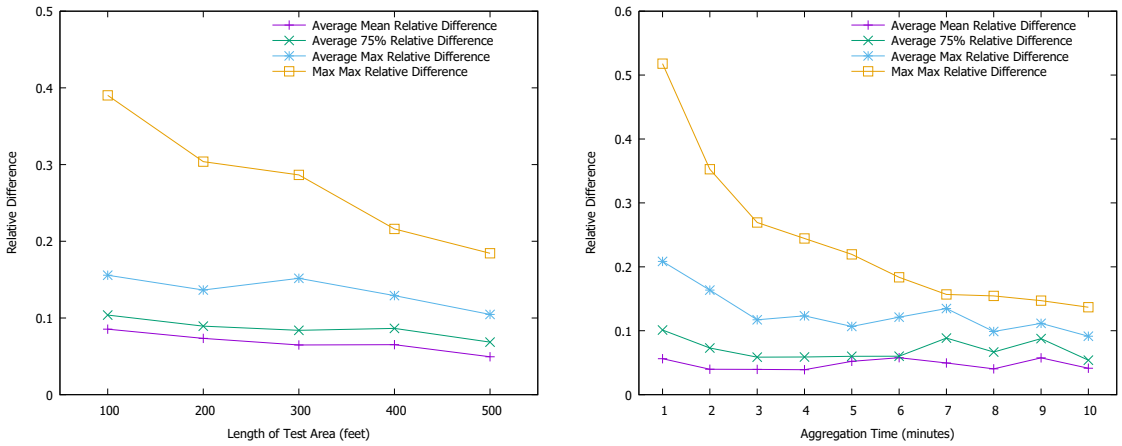


(e) NGSIM Lankersim 50% penetration over two minutes



(f) NGSIM Lankershim Result Summary

Fig. 56: Results for NGSIM traffic.



(a) NGSIM Lankersim Sensitivity by Length (b) NGSIM Lankersim Sensitivity by time

Fig. 57: Sensitivity plots

decreases as the penetration rate increases. This is also why the Average Mean result is always less than the actual value and typically the maximum in value for the results is actually closest to the actual value.

6.1.10 SENSITIVITY ANALYSES

In this section we provide sensitivity analysis to our method in two ways. First we vary the length of the test area to test if the method will work with shorter roadways. Second, we vary the time span of the increments to see how the method compares with more and less granularity.

Sensitivity analysis by length

To conduct a sensitivity analysis by the length of the test area, the simulation is modified to vary the distance of the roadway being considered. The start of the test area and the location the flow is being measured remains fixed from the southend. From this fixed point, a test area of 100, 200, 300, 400, and 500 feet are considered and reported in Figure 57(a). A penetration rate of 50% is considered for each of the runs. The simulation for each of the five lengths is run ten times. The results are aggregated the same as the penetration rate summary, except that the results are aggregated every 5 minutes. First the results

are aggregated per time period, then the *Average Mean Relative Difference*, *Average 75% Relative Difference*, and *Average Max Relative Difference* are calculated and plotted. The *Max Max Relative Difference* is also included show the maximum relative difference across all the runs.

The results interestingly show that even at 100 feet the results are still quite good. This is unexpected, but it can be explained by considering gaps in the results. The method aggregates the counts of vehicles between several sequential southbound vehicles involved in the method. As long as there are no gaps, the length of the test area will not matter. The difference in the measured and the actual value vary because of the gaps present in the results. These results show that there are only a few more gaps in the results at 100 feet than at 500 feet. As the length of the test area increases, the number of gaps in the results decreases, leading to an increased accuracy of the result.

The *Average Mean Relative Difference* decrease slightly between 8.5% with a 100 foot test area to 4.9% with a 500 foot test area. The *Average 75% Relative Difference* also decreases slightly, with 10.4% a 100 foot test area to 6.8% with a 500 foot test area. The *Average Max Relative Difference* decreases similarly, starting with 15.6% with a 100 foot test area to 10.5% with a 500 foot test area. Finally, the *Max Max Relative Difference*, which shows the maximum difference for each run starts with a relative difference of 39.0% with a 100 foot test area down to 18.4% with a 500 foot test area.

Sensitivity analysis by time

An additional sensitivity analysis is provided looking at changing the time span the results are aggregated. For this, the same results as in Subsection 6.1.8 for the penetration rate of 50% is used, but aggregated over several time spans; each minute increment from one to ten. Again, as before, each of the results are aggregated for each of the varying time periods, then the *Average Mean Relative Difference*, *Average 75% Relative Difference*, and *Average Max Relative Difference* are determined and plotted. Again the *Max Max Relative Difference* is included show the maximum relative difference across all the runs.

The *Average Mean Relative Difference* is relatively stable between 5.7% with a one minute aggregation time to 4.1% with a ten minute aggregation time. The *Average 75% Relative Difference* decreases slightly, with 10.1% with a one minute aggregation time to 5.4% with a ten minute aggregation time. The *Average Max Relative Difference* is more interesting, starting with 22.5% with a one minute aggregation time down to 9.4% with a ten minute aggregation time. Finally, the The *Max Max Relative Difference*, which shows

the maximum difference for each run starts with a relative difference of 51.8% with a one minute aggregation time down to 13.7% with a ten minute aggregation time.

6.1.11 COMPARISON WITH THE MO METHODS

The MO3 method is designed for urban roadways; however, as described the results were inaccurate given the highly variable traffic flow. The method is expected to produce results that rival MO1 and MO2 for highways, but given the requirement that oncoming vehicles must be countable, and in this work, by radar, this method is not comparable to the other methods.

The MO3-flow method, like MO1, will produce a result for traffic flow; however, a key difference is the MO1 method is designed for long highway roads where the MO3-Flow method is designed for urban roads. As such, the methods are not comparable.

6.1.12 THE MO3-FLOW METHOD IN PRACTICE

In the MO3-Flow method there is a mix of enabled and non-enabled vehicles. The method uses enabled vehicles to account for non-enabled vehicles in both codirectional and oncoming traffic. Additionally, an enabled vehicle must have the ability to communicate wirelessly with other enabled vehicles and establish an agreement to run the MO3-Flow method. This method is designed to work on short urban road segments, for example between two intersections, where there is no occlusion between the two directions of traffic. There should be no additional entrances or exits.

We now describe how the MO3-Flow method will work using the messages and formats described in Section 3.3.

A TMC or local authority will establish a list of roadways that it requires flow approximations using the MO3-Flow method. It will assign a Test Area ID to each and establish the geometry of the roadway. It will then update RSUs or local agents near the test areas with the list of test areas.

The RSU or local agent will then regularly broadcast a WSA message using the established PSID for the test area advertisement service. The WSA will include details of the channel and details to receive WSM messages associated with test area advertisements.

The RSU or local agent will regularly broadcast the locations of nearby or upcoming test areas so vehicles within communication range are aware of the locations. This is done through the use of the Test Area Advertisement (TAA) messages described in Section 3.3.1. The TAA message will be sent using the WSM format and will contain the Test Area Id,

the MO Method Type for the test area, and the geometry describing the boundaries of the test area.

The enabled vehicles will read the messages and locate the test area geometry on its digital map. Then, using its GPS and digital map, the vehicle will determine the moment it enters into a test area.

When it does enter the test area, it will begin maintaining separate tallies codirectional and oncoming traffic. For codirectional traffic, the tally is the number of times a vehicle passes it minus the number of times it passes another vehicle. For oncoming traffic, the tally will be a count of the number of vehicles it meets.

Also upon entering a test area, the vehicle will advertise itself as an enabled vehicle. It will start the MO Partner Negotiation by sending a MO Partner Request (MPR) message, as described in Section 3.3.3. This message is a broadcast message sent using UPD over IPV6. This message will include the Test Area Id, the MO Method Type, and its own Temporary Id. Note that the Temporary ID used in the MPR message is the same that is used in the BSM message; this ensures any vehicles receiving the MPR can determine the location of the sender. The MPR message is sent regularly throughout the test area to ensure it can establish a MO Partner with any possible enabled vehicles.

When an enabled vehicle receives an MPR, it will check if it is within the same Test Area that is sent in the MPR message. If it is, it will then determine the location, velocity, and heading of the sender from a BSM message with the same Temporary ID as the MPR message. If both are traveling in the opposite direction toward one another, it will respond to the MPR with a MO Partner Accept (MPA) message. This message is described in Section 3.3.4. The vehicle receiving the MPR message knows the IPV6 address of the sender from the From Address of the UDP message. When responding it will send the MPR message using TCP over IPV6 and will send it to this same From Address. This message will include the Test Area Id, Start Time, MO Method Type, and its own Temporary Id. The receipt and acknowledgment of the MPR message establishes that they are both part of an MO Partnership which starts at the Start Time. This Start Time will be the time later time between the Start Time of the MPR message and the time the vehicle entered the test area. The Start Time from the MPA message will be the time used in the MO3 Method.

During the MO Partner Negotiation, each vehicle sends to the other its Temporary Id. Given that BSM messages are send regularly, each MO Partner will track the location, velocity, and heading of the other partner. Specifically, each MO Partner will store the the IPV6 address, Temporary Id, and Start Time. Each time it receives a BSM from the

Temporary ID of its partner it will update the location and the distance between the two of them.

Each partner will track the other using the BSM messages. Each will regularly estimate the time they will pass. Note that this does not need to be done with every BSM message received; however, as the two get closer and closer, it will need to estimate the time they pass more and more to establish the exact moment they pass.

Upon the moment they pass, the vehicle will create a Tally Exchange Message (TEM) as described in Section 3.3.5. This message marks the start of the first phase of MO3-Flow.

The TEM will include the tally information from the first vehicle. The following fields are included:

- Test Area ID - This is used to verify the test area being measured.
- Start Time - This is the time the vehicle entered the test area.
- End Time - This is the time that the vehicle determined as the time of passing.
- Start Location - This is the starting location of the vehicle.
- End Location - This is the final location of the vehicle at the time the vehicles passed
- Heading - This represents the heading of the vehicle.
- Tally Type - The tally type will be a 0 to represent that this is the first phase of the MO3-Flow method.
- Tally Codirectional - This is the codirectional tally of the vehicle.
- Tally Oncoming - This is the oncoming tally of the vehicle.
- Flow - This field is left blank for the MO3-Flow method.
- Density - This field is left blank for the MO3-Flow method.

The TEM message is sent using TCP over IPV6 with the respective IPV6 To and From address fields filled appropriately. After retrieving the message from its MO partner, the partner will start the second phase of the method. It will save the information from the TEM message and will start a new tally and maintain it until it crosses out of the test area.

Upon exiting the test area, the partner vehicle will create a Traffic Parameter Collection (TPC) message to send to the next RSU advertising the TPC service. The vehicle will listen

for WSA messages that advertise the PSID associated to the TPC service. Once it receives one, it will create the TPC message and send it to the RSU via a WSM. The TPC message includes the following fields:

- Test Area ID - The Test Area ID identifies which Test Area the MO Measurement data is being reported.
- MO Method Type - The MO Method Type serves two purposes. First it is used to verify the method used, and second, it is used to determine the format of the MO Measurement Data.
- Start Location - The Start Location is the first field of the MO Measurement Data. It includes the location data of the vehicle at the time of the density measurement.
- End Location - The End Location field is the second field of the MO Measurement Data. It includes the location data of the vehicle's MO partner vehicle at the time of the density measurement.
- Time - The Time field is the third field of the MO Measurement Data. It is the time of the density measurement.
- Density - The Density field is the fourth and final field of the MO Measurement Data. It is the density between the Start Location and End Location at the time represented in the Time field.

The MO3 method is complete when the second vehicle exits the test area and the TPC message has been sent.

6.2 SUMMARY

In this chapter, the MO3 and MO3-Flow methods were introduced. The contribution of this chapter is to show that vehicles in oncoming traffic can be used to determine the density and flow. In comparison to the MO1 and MO2 methods, both MO3 method uses the same idea of the tally; however, here we exploit an assumption that as two oncoming vehicles meet, the number of vehicles between them is assumed to be zero. We started with the hypothesis that using oncoming traffic will allow the method to work in urban environments. We have found that using the MO3 method to measure traffic in urban environments suffered from some issues; however, it did allow density to be measured in a

highway setting within a short distance. From the issues in the MO3 method, the idea was born to measure the flow, and to aggregate the counts in a unique way to mitigate these issues.

Extensive simulations were performed using actual vehicle traces using the HighD and NGSIM Lankershim datasets. The result have shown the MO3 density method can be used in highway and suburban traffic settings and the MO3-Flow method can be used to determine the flow in urban traffic where the flow of vehicles changes rapidly. The advantages of both methods include the simplicity of the methods as well as they both only need V2V communications to collect the data. Additionally, as with MO1 and MO2, the MO3 and MO3-Flow methods are privacy-preserving,

In the previous chapters and especially with the MO3-Flow method, we suggest that an RSU or local agent is used to collect the TPC messages. In the next chapter we show how this local agent may be one of the vehicles on the roadway itself!

CHAPTER 7

VIRTUAL ROAD SIDE UNIT

7.1 INTRODUCTION

In the MO1, MO2, and MO3 Methods, there is no reliance on the RSU to measure the traffic parameters; however, in MO3-Flow, the RSU is needed to aggregate the data. This means there is a dependency on where the MO3-Flow method can be used. It must be used near an existing RSU. It is typically seen that there is an RSU that acts as the main point of communication within a particular service area, that is the roadway within radio range of the RSU. Similarly, in V2X, the host of a service is typically an RSU; however, vehicles are not disqualified from hosting a service [47].

The basic idea of this chapter is to supplement the RSU with a service running on a vehicle within a particular service area. This service will serve the same purpose as a RSU, but instead will be running on a vehicle. As such, we call this a Virtual Road Side Unit (VRSU). As the vehicle nears the boundary of the service area, it will begin a migration process where the data is migrated to another vehicle. The goals of this migration is for the service and data remain in the service area for as long as possible and to reduce the down time of the service.

The main contribution of this chapter is to provide a VRSU architecture for running a reliable virtual service using unreliable vehicular traffic.

7.2 VRSU ARCHITECTURE

7.2.1 VEHICLE MODEL

As is the case today, we expect vehicles in the future will have different capabilities; in this chapter we assume all enabled vehicles meet a base set of requirements. These requirements include:

- the ability to communicate with infrastructure and other vehicles using a V2X compliant radio,

- the ability to track its own location via GPS,
- the ability to pinpoint its own location and other vehicles' locations on a digital map,
- the ability to run a service image using a lightweight virtualized container.

The original intention is to use the vehicle's unused processing power from the vehicles main CPU to run services. Given the importance of the vehicle's main CPU for driving and passenger safety, and alternative idea is for the vehicle to have a second dedicated CPU to handle non-critical services and application.

The virtualization architecture for the service is split into three parts. First, there is a base operating system of the vehicle. This operating system is able to run multiple virtualized containers, such as that provided by Linux Containers. The second part is the Service Image which contains the code and libraries required for the service to run. Finally, the third part is a separate Data Image where the data of the Service Image is saved.

The vehicle is preinstalled with the operating system. Additionally the Service Image may either be preinstalled or downloaded from an RSU. Since each enabled vehicle has the operating system and the Service Image already, when the service is migrated to another host, only the Data Image needs to be migrated. This ensures the data being sent from host to host is minimized.

The intention of the VRSUs is to be used to provide supplemental mini services throughout the roadway that do not require large datasets. The benefit is that many can be provisioned without requiring hardwired infrastructure. In this chapter, we will consider the VRSUs as a service that is providing data collection and aggregation for the MO methods, specifically, providing aggregation support for the MO3-Flow method.

7.2.2 INFRASTRUCTURE MODEL

In this architecture, there are RSUs available on the roadway; however, these are supplemented by Virtual RSUs (VRSUs) which are able to bring data collection and aggregation closer to the MO test areas without needing to provide a hardwired RSU. The RSUs will have wired network access connected to the Traffic Management Center (TMC); however, the VSRUs will not have any wired network access. See Figure 58 for a visual representation of this model.

In the MO methods, the RSU has two responsibilities. The first is to notify the vehicles of the locations of test areas through the use of Test Area Advertisement (TAA) messages. The second is to run the Traffic Parameter Collection (TPC) message collection service to

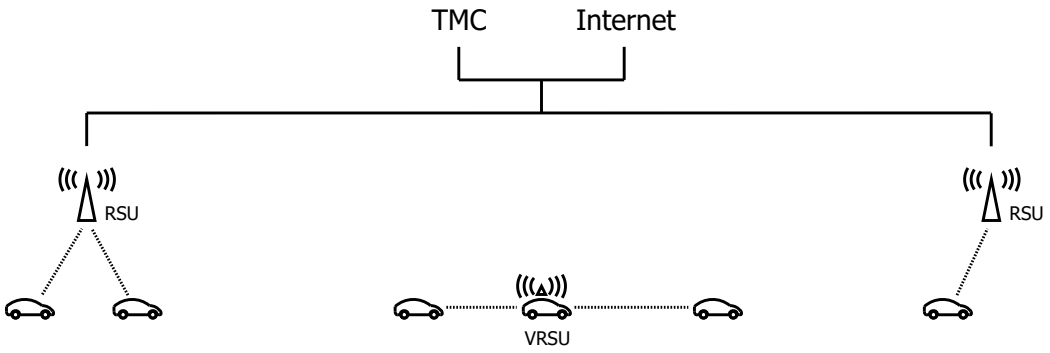


Fig. 58: VRSU architecture

collect TPC messages from vehicles. The RSUs will continue to handle these responsibilities; however, it will also hold one additional responsibility. Given the RSUs have wired network access to the TMC and the VSRUs do not, the VRSU must have the ability to send data to the TMC through the RSU. This is explained later in the chapter in Section 7.2.5.

7.2.3 MESSAGES SENT

In the VRSU architecture, it is assumed there are some strategically placed RSUs that are regularly updated with the latest Test Areas by the TMC. Given that VRSUs are not directly connected to the TMC and cannot be updated in real-time, we assume the TAA messages will only be sent by RSUs.

The TPC message collection service will be run on both RSUs and VRSUs. The unique feature of the VRSU is that it can be provisioned anywhere on the roadway. One can be placed near a MO test area to ensure the results are collected and aggregated quickly.

To setup the VRSU and to handle migration, there are four additional messages that will be sent: The first is the VRSU Location Advertisement which is sent by the RSU to notify vehicles of the locations of VRSUs. The second is the VRSU Advertisement in which is sent by the host vehicle will advertise itself as the VRSU host. It is also used to advertise the next host. The third is the VRSU Registration which is used by vehicles to register to be a candidate for the next VRSU host. The final message is the VRSU Migration which is sent by the VRSU host to migrate data to the next host.

VRSU Location Advertisement

In this architecture, the RSUs will advertise the locations of the VRSUs. This is beneficial because the TMC may provision a VRSU on the fly. More of this provisioning is explained later in the chapter in Section 7.2.4.

The VRSU Location Advertisement (VLA) message is provided by RSUs to advertise the locations of VRSUs.

Just like the TAA message, this advertisement is a good candidate for the WSA and WSM messages. An RSU or a group of RSU's will each provide a service for advertising the local VRSU service areas and parameters. Each RSU will include in their WSA message the advertisement for the VLA service, which includes the Provider Service IDentifier (PSID) representing the VLA service. Enabled vehicles will receive the WSA message and will subscribe to the service. Then the RSU will regularly send WSM messages with a list of VRSU Location Advertisements

TABLE 11: The VLA Format

Field	size
VRSU Service Area ID	2 bytes
VRSU Origin	8 bytes
Geometry	298 - 1194 bits

The Payload of the WSM data field includes a list of VLAs with the following format that is also shown in Table 11:

VRSU Service Area ID - The VRSU Service Area ID is a unique identifier for the VRSU service area. This is used by other messages to identify the Service Area. It is 2 bytes long giving a total number of 65536 possible Ids.

VRSU Origin Point - The VRSU Origin Point represents a point that will be used as the origin point of the VRSU. This origin point will typically be the center of the service area as designated by the Geometry. It uses the Position2D field type which includes the latitude and longitude values. The precision of each field is 1/8th micro degrees. Both fields

together use 8 bytes.

Geometry - The Geometry field contains subfields that give identifying information to the road as well as descriptions of the start and end boundaries of the service area. The description of the Geometry section will be described below. We note here that the minimum size of the Geometry field is 298 bits and the maximum size is 1194 bits.

This completes the VLA message format.

The Geometry Field from the VLA message is the same as that of the TAA message in Chapter 3. It is duplicated below in full detail, as well as in Table 12(a):

Road ID - The Road ID field is a 4 byte unique identifier that represents a road.

Heading - The Heading field is a 2 bit value that represents the direction of travel of the roadway from the Boundary Start line to the Boundary End Line. This field has the following assignments:

1. North,
2. South,
3. East, and
4. West.

Boundary Start - The Boundary Start field represents a line that marks the starting line of the service area. It is represented with a Boundary Type format. The Boundary format will be described below. We note the minimum size of the boundary field is 132 bits, and the maximum size is 580 bits.

Boundary End - The Boundary End field represents a line that marks the end line of the service area. It is also represented with a Boundary Type format.

This completes the Geometry Field format.

The Boundary Field from the Geometry field is now described in full detail, as well as in Table 12(b):

Reference Point - The Reference Point represents a point that will be used as a reference point to draw a boundary line using multiple nodes. It uses the Position2D field type which includes the latitude and longitude values. The precision of each field is 1/8th micro degrees. Both fields together use 8 bytes.

Number of Nodes - The Number of Nodes field contains an integer value representing the number of nodes in the Node List field. This field has a size of 4 bits, which represents a maximum of 16 nodes in the node list. There must be at least 2 nodes in the list.

Node List - The Node List field contains a variable number of node fields. Each node field contains an x and y offset represented in centimeters from the Reference Point. Each node is ordered and are used to draw a line from one side of the road to the other designating either the start or end line of the boundary. The x and y offset fields are both each 2 bytes and represent a signed integer value between -32,768 ad 32,767. The number of node fields is set as the Number of Nodes field which can be between 2 and 16. This means the minimum size of the Node List is 64 bits, or 8 bytes. Also, the maximum size of the Node List of 512 bits, or 64 bytes.

This completes the Boundary Field format.

TABLE 12: Geometry and Boundary Format

(a) The Geometry Format

Field	size
Road ID	4 bytes
Heading	2 bits
Boundary Start	132 - 580 bits
Boundary End	132 - 580 bits

(b) The Boundary Format

Field	size
Reference Point	8 bytes
Number of Nodes	1 byte
Node List	8 - 64 bytes

VRSU Advertisement

The VRSU Advertisement (VA) message serves two purposes. First it advertises the current status of the VRSU. Also it notifies the existing vehicles in the service area if a migration is in process to a new host and which vehicle will become the host after the migration is complete.

This advertisement is a candidate for using WSA and WSM messages. The VRSU sends a WSA message to advertise itself as a VRSU and, if migrating, notifys other vehicles of the next host. The message includes the Provider Service IDentifier (PSID) representing the VA service. Additionally, the Service Info of the WSA includes the IPv6 address and Service Port of the current host which is important for the next message. Enabled vehicles receive the WSA message and then subscribe to the service. The the VRSU regularly sends

WSM messages with additional information.

TABLE 13: The VA Format

Field	size
VRSU Service Area ID	2 bytes
Host Temporary ID	4 bytes
Status	2 bits
Next Host Temporary ID	4 bytes

The Payload of the WSM data field includes VRSU host information with the following format that is also shown in Table 13:

VRSU Service Area ID - The VRSU Service Area ID is the ID that represents the Service Area ID of the VRSU. This helps the listener of the message to match the advertisement with the VLS message.

Host Temporary ID - The Host Temporary ID is the ID the host vehicle will use throughout the service area when sending BSM messages. This ensures the vehicles within the service area are aware of which vehicle is the VRSU and is able to track its location using BSM messages. The Temporary ID field is represented by 4 bytes.

Status - The Status field identifies the VRSU as initializing, active, migrating, or failed. If the service has just started and not yet active, the status will be *initializing*. When the service is running and is in its service area, the status will be *active*. If the service is in its service area, but the host is being migrated to a new host, then the status will be *migrating*. Finally, if the service is outside of its service area, and it has not successfully migrated, the status will be *failed*. The field is represented with 2 bits; the values are:

1. initializing,
2. active,
3. migrating, and

4. failed.

Next Host Temporary ID - The Next Host Temporary ID is the ID the next host vehicle will use throughout the service area when sending BSM messages. This ensures the vehicles within the service area are aware of the next host and can track its location using BSM messages. If the status of the VRSU is active, then the Next Host Temporary ID may be blank. If it is active, the next host may already have been chosen. The Temporary ID field is represented by 4 bytes.

This completes the VA message format.

VRSU Registration

The VRSU Registration (VR) message is used for vehicles within the service area of the VRSU to register as a candidate for migration. For the service to remain in the service area, it must be migrated often; it is important for candidate vehicles to register so the most ideal vehicle can be chosen as the next host.

In the VA message, the Service Info of the VRSU host includes the IPv6 Address and Service Port. The candidate vehicle will send a registration message to this address and port using TCP over IPv6.

TABLE 14: The VR Format

Field	size
Candidate Temporary ID	4 bytes
Merge Flag	1 bit

The Payload of the TCP message includes the following format that is also shown in Table 14:

Candidate Temporary ID - The Candidate Temporary ID is the ID the vehicle will use throughout the service area when sending BSM messages. The host will use this Candidate Temporary ID to track the candidates' locations using BSM messages. The Temporary ID field is represented by 4 bytes.

Merge Flag - The Merge Flag field is used to indicate to the VRSU host that it has data to merge into the current host's Data Image.

This completes the VR message format. Note that the TCP header will include the IPv6 from address of the candidate. Both are important information in choosing a new candidate host.

VRSU Data Migration

When the next host has been chosen, a single VRSU Data Migration (VDM) message is sent to the new host that contains all the data within the Data Image of the VSRU service. The data is sent using TCP over IPV6. TCP is used for this message because it is a reliable protocol and also because TCP can be used to segment a large message into smaller messages that can be sent to and reordered properly at the destination.

TABLE 15: The VDM Format

Field	size
Merge Bit	1 bit
Data	variable bits

The Payload of the TCP message includes the following fields that are also shown in Table 15:

Merge Bit - The Merge Bit is used to determine if the data from the message should be merged into the hosts existing Data Image. This bit is set when a Data Image from a failed VRSU migration is being sent back to the VRSU. This is explained in more detail in Section 7.2.5.

Data - The Data being sent is the bits from the Data Image to be migrated.

This completes the VDM message format.

7.2.4 PROVISIONING A NEW VRSU

We now look at how a new VRSU is started and also how it is restarted in case of a migration failure. To start, we must rely on the RSU infrastructure to provide vehicles with the VRSU origin locations and the service areas for each of the VRSUs; this is done using VLA messages.

When a vehicle enters the VRSU service area, it expects to receive a VRSU Advertisment. If it does not receive one by the time it reaches the VRSU origin, then it will appoint itself as the VRSU initializer. It will send out a VRSU Advertisement with a status of *initializing*. When in this status, other vehicles will still register with the VRSU; however, the VRSU service is not yet active.

In the time the VRSU is first initiated and the time it must migrate, the VRSU will receive a number of VRSU registration messages. If it does not receive any, then the VRSU will stop sending messages and will terminate the service. If it does receive messages, then the most ideal candidate will be chosen, and the migration will start. The vehicle will send a zero sized VRSU Data Migration message to the candidate, and the candidate will take over as the new VRSU. The new VRSU host will send VRSU Advertisements with the status as *Active*.

7.2.5 MIGRATION

Given vehicles move on the roadway, using them as the infrastructure requires the data from a host to be migrated to a new host vehicle. To make the migration seamless, the amount of time spent migrating must be reduce and the roles of the current and next host must be defined.

As mentioned previously, the architecture of the service is divided into three parts, of which only the data needs to be migrated. This is done intentionally to minimize the amount of data that needs to be migrated. Given the restricted bandwidth in a vehicular network, reducing the amount of data to migrate is extremely important.

First we will discuss the role of the host, then we give more details into the size of the Data Image.

Role of the host

To ensure a seamless migration, we must define the roles of the current host and the next host. Given that only the Data Image is being migrated, and the service will be running on both the current host and the next host; this can be considered a live migration. Any new TPC messages being sent to the VRSU will be received by both hosts. In this work, there is

no means to query the VRSU for its information; however, the current host will handle this role in future work involving the VRSU. Since the current host has all of the data image, it is the proper host to answer any queries.

Upon the Data Migration message completing, the current host will cede the responsibility of hosting the VRSU to the next host. The current host will assume a Post Migration role, see Section 7.2.5. The next host will become the current host and will start sending the TPC WSA messages as well as the VA messages.

Data Image Size

Given this architecture is developed to aid the MO3-Flow method, it is important to understand the data needs of the MO3-Flow method. In the MO3-Flow simulation in the previous chapter, the number of measurements was also recorded over the 30 minute simulation periods. Over the ten runs for each penetration rate tested, the minimum, 25% quartile, average, 75% quartile, and maximum measurements counts for the northbound traffic are plotted in Figure 59. If the southbound traffic is also included, then the values can each be doubled to determine the total possible number of messages the VRSU will receive. In the case of the 50% penetration rate, the maximum value is roughly 6000. Assuming a similar southbound traffic, the number of messages will be doubled to 12000.

The size of each TPC message, according to Chapter 3 is 12.5 bytes. Given the maximum value at 50% penetration is 12000, there will be a total of 150 kilobytes of data generated in a 30 minute window. Granted, every message will not need to be saved. An additional role of the VRSU is to aggregate the data regularly. In this chapter, we assume this will be done at least every 15 minutes, but perhaps more often. After the data is aggregated, there is little need to keep the raw TPC messages. This aggregation and cleanup will keep the Data Image size quite small.

When to migrate

See Figure 60 for a visual representation of the concepts discussed in this section.

Migration of the VRSU is important to ensure the continuity of service in the VRSU service area. As the hosting vehicle passes by the VRSU origin, it will start to travel away from the origin. Given that migrations take time to complete, prior to reaching the boundary of the service area, it will need to start a migration. This area is called the Migration Zone.

The minimum distance to complete a migration, d_m, is given in Equation (48):

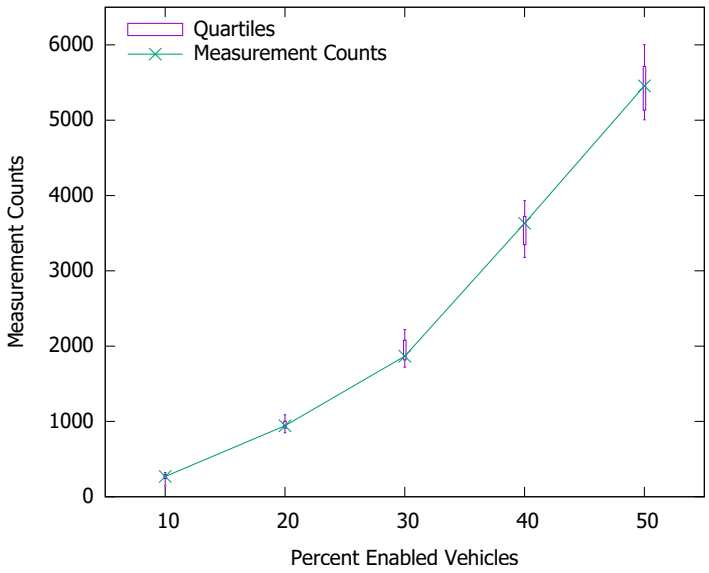


Fig. 59: MO3 measurement counts

$$d_m = v \cdot \frac{s}{b} \tag{48}$$

where v is the velocity of the host vehicle, s is the size of the Data Image, and b is the bandwidth of the network. d_m represents the minimum possible size of the migration distance; however, since continuity of service is important, this distance will be increased by a certain factor, f, to take into account possible network issues. When the host vehicles travels within a distance $d_m \cdot f$ from the boundary of the VRSU Service Area, the recruitment and migration of the VRSU Data Image must have already began, or a migration failure is likely to occur.

Recruitment

After it is determined that a migration is required, candidates are placed into two sets to determine which the next host. First, to be a candidate, the vehicle must be within communication range of the current host. Next, the candidates are placed in two sets based on its heading, which is determined from BSM messages.

- The first priority set is vehicles that are moving toward the VRSU origin. These vehicles may be codirectional or oncoming vehicles. This priority set is sorted by the

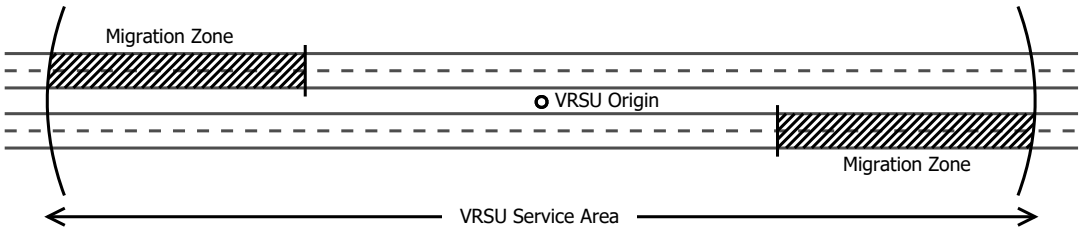


Fig. 60: VRSU migration

vehicle's distance to the VRSU origin in descending order. The vehicle that is the farthest from the VRSU origin that is heading towards it is expected to remain in the service area the longest.

- The second priority set is vehicles that are moving away from the VRSU origin. Again, these may be codirectional or oncoming vehicles. This priority set is sorted again by the vehicle's distance to the VRSU origin, but in ascending order. In this priority set, it is the vehicle that is closest to the VRSU origin but moving away from it that is expected to remain in the service area the longest. Additionally, the candidate must not be farther away from the VRSU origin than the current host.

Once the candidates are sorted into the two priority sets, the candidate is chosen as follows. If there is least one candidate in the first priority set, use the vehicle that is the farthest from the VRSU origin. If there are no candidates in the first priority set, then use the vehicle in the second priority set that is the closest to the VRSU origin. If there are no candidates in either the first or second priority set, then wait until a valid candidate registers to migrate. If a valid candidate does not register while the host is in the service area, then this is considered a failed migration. This scenario is covered in the next section.

Failed Migration

If the VRSU does not complete the migration until after it leaves the VRSU Service Area, this is considered a failed migration.

The host will remain active, but in a failed migration status. It will find the first suitable host traveling in the opposite direction, which is the direction towards the service area of the

VRSU. The goal of this migration is to get the Data Image back to the service area. In the meantime, it is likely that a new VRSU has declared itself the new host and is initializing a new VRSU. The new VRSU will choose a new host and the VRSU will continue as an active service.

As mentioned previously, the failed host of the VRSU must find the first suitable host traveling in the opposite direction. The failed host will continue to send VA messages with a Status of *failed*. Enabled vehicles will send VR messages to the failed host to register with it. Upon receiving a VR message from a suitable host, the migration process will start and a VDM message is sent with the Data Image.

This new host will now travel back towards the VRSU Service Area. Nearing the service area, it may hear a VA message advertising the VRSU for the service area.

If the host does not hear a VA message advertising the VRSU by the time it reaches the VRSU Origin, it will declare itself as the new host and send a VA message itself. After choosing a new candidate, it will migrate the Data Image it received from the failed host instead of an empty Data Image.

If the host does hear a VA message, it will register with the active VRSU using a VR message. The VR message format includes the Merge Flag which will be set to 1 to indicate the candidate has data to merge. The active VRSU host will verify if it can migrate it's Data Image to the candidate holding the Data Image to merge. If it can, it will migrate to the candidate and the candidate will run a process to merge both sets of data together.

The Data Image from the failed migration has now been returned to the VRSU Service Area.

Of course it is certainly possible that one of the steps to return the Data Image to the VRSU may fail. The failed host will make one attempt to migrate the Data Image to a new host traveling in the opposite direction. Also, when entering the VRSU service area, the candidate holding the Data Image may not be chosen as the next VRSU host. If either of these occurs, the Data Image merge will be considered failed. The VRSU will enter into an inactive status and will begin Post Migration responsibilities.

Post Migration

After the migration to a new host is complete, or if the Data Image merge has failed, the VRSU will remain in an inactive state. It will run a process on the Data Image that will create a summary of the data to send to an RSU, which will then be sent to the TMC.

The TPC Summary message is created by aggregating each of the individual TPC messages and creating a summary of the data. To send the message, we rely strictly on data muling to deliver the message from the host to the RSU. Upon reaching an RSU that is advertising the TPC collection service, it will send the RSU its aggregated data. This is done by the use of the TPC messages by setting the Summary Bit in the MO Method Type - Extension field to a 1 to indicate it is a Summary message.

One alternative to relying on data muling only would be to use a routing protocol such as OPERA that is proposed by Abuelela *et al.* [127]. OPERA, or Opportunistic Packet Relaying protocol, is a routing protocol for Delay Tolerant Networks that combines data muling and local routing. The authors' goals in the protocol are to minimize the delivery time and the hop count while routing in a disconnected vehicular network.

If data muling alone is considered, then the vehicle will hold onto the data and deliver it to the RSU when the vehicle comes within transmission range. This will minimize the hop count to one; however, the delivery time will not be minimized.

The basic idea of OPERA is to route the message opportunistically. If the origin vehicle can send the message to another codirectional vehicle in front of it, it will do so until there are no more forward vehicles within range. If the origin vehicle cannot send to another codirectional vehicle in front of it, it will find a suitable vehicle in the oncoming traffic. The suitable oncoming vehicle must be able to send the message to another vehicle in front of the origin vehicle but is out of range of the origin vehicle or route the message to another vehicle behind it that is able to.

Once the data is sent to the RSU using data muling or routed opportunistically using OPERA, the VRSU service and accompanying Data Image are no longer necessary. The Data Image will be deleted from the vehicle and the VRSU service will be terminated.

7.3 VRSU SIMULATION AND RESULTS

7.3.1 SIMULATION MODEL

The VRSU method was tested with a simulated roadway traffic dataset that was simulated using SUMO [125]. The roadway has two lanes going in both directions and includes no on-ramps or off-ramps other than the start and end of the road. Vehicular traffic in both directions have the same flow and traffic will run for three hours.

The VRSU origin is placed in the middle of the lanes with the service area stretching 1km in either direction from the origin. The VRSU must stay within the service area, or

a new VRSU will be chosen. Each of the experiments run consider the metric *Number of VRSU Births*. This is the number of times the VRSU is restarted. Each experiment will start with one VRSU Birth. Each time there is a failed migration, a new VRSU will be spawned and the Number of VRSU Births will be incremented. Note that the experiment is only testing the ability to maintain the VRSU using migration only. Once the VRSU is failed, the process to merge it back into the active VRSU is not being performed in the simulation.

The vehicle will begin migration 500 meters from the boundary of the service area, or sooner. Using Equation (48), the minimum distance to migration depends on the velocity of the vehicles, the bandwidth available, and the size of the data image. The greater value of $d_m * 1.15$ and 500 is used, where 1.15 is a factor taking in account possible networking issue. Vehicles are asumed to have V2X compliant radios and can communication up to 1km. This is why 500 meters is chosen, as this gives equal consideration to selecting codirectional and oncoming vehicles as candidates for migration.

Throughout the experiments, four values are varied: vehicle velocity, traffic flow, network bandwidth, and Data Image size.

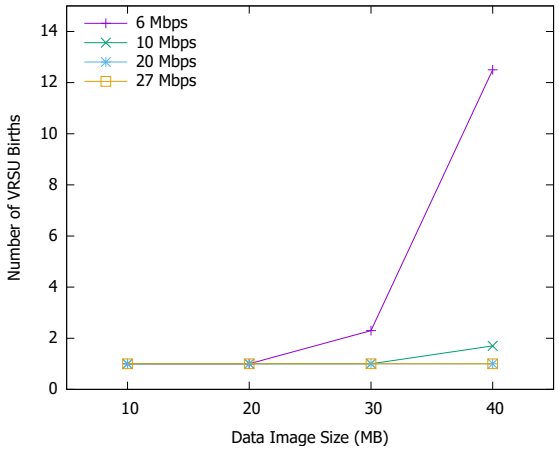
Two values for vehicle velocity is used in the experiments to offer comparable simulations of urban and highway traffic. The urban traffic is simulated using traffic traveling at 35 mph (about 55 km per hour). The highway traffic is simulated using traffic traveling at 55 mph (about 88.5 km per hour).

Three different flows, 500 vehicles per hour, 1000 vehicles per hour, and 2000 vehicles per hour, are used to distinguish between low, medium, and high flow traffic.

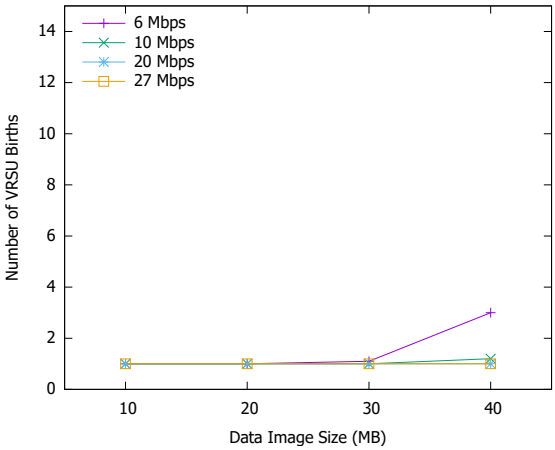
The network bandwidth is varied between 6 Mbps, 10 Mbps, 20 Mbps, and 27 Mpbs. Note that the units are in Mbps which is Mega bits per second.

The Data Image Size mentioned previously in the chapter had a maximum value of 150 kilobytes for the MO3-Flow TPC messages. Granted, this chapter is written with MO3-Flow in mind, but the idea can be extended to other services as well. To fully understand the conditions in which the VRSU architecture will work, we vary the Data Image Size between 10, 20, 30, and 40 Megabytes.

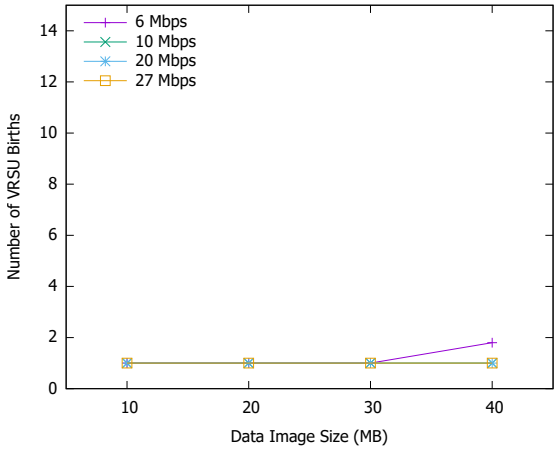
Finally, the last parameter of the simulation is the percentage of enabled vehicles. In the experiments, we keep the parameter set to .5, meaning 50% of the vehicles are considered enabled. We offer additional sensitivity analysis to this parameter in Section 7.3.3



(a) VRSU births for 500 vehicles per hour.



(b) VRSU births for 1000 vehicles per hour.

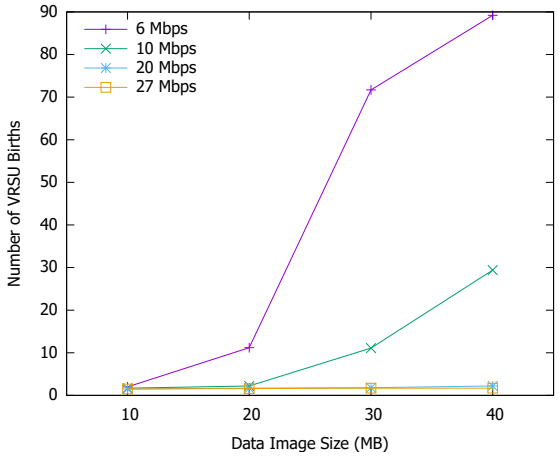


(c) VRSU births for 2000 vehicles per hour.

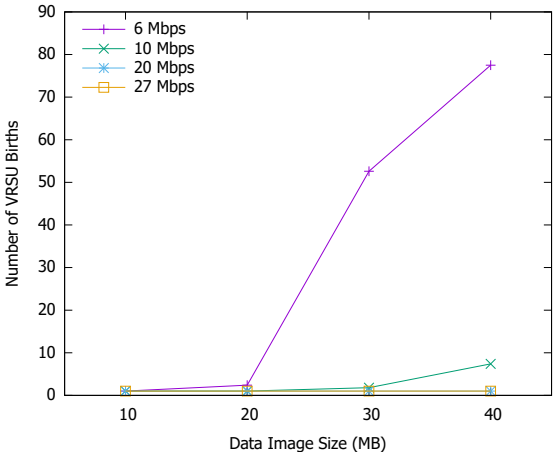
Fig. 61: The number of VRSU births over three hours for traffic driving at 35mph.

7.3.2 SIMULATION RESULTS

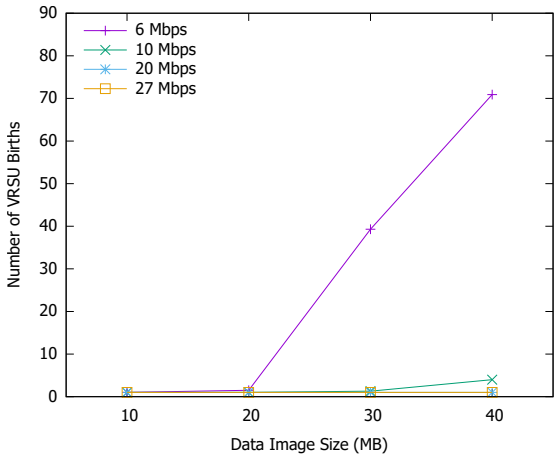
The simulation results are split into two sets based on the velocity of the vehicles. Both sets include experiments with the bandwidth and Data Image size varying. The bandwidth varies between 6, 10, 20, and 27 Mbps. The Data Image size varies between 10, 20, 30, and 40 MB. For each configuration, the simulation is run ten times and the Number of VRSU Births for each run is averaged to get the value that is included in the plot. In both Figure 61 and Figure 61, the plots are organized with the y-axis representing the Number of VRSU Births, the x-axis representing the Data Image Size, and the plots representing



(a) VRSU births for 500 vehicles per hour.



(b) VRSU births for 1000 vehicles per hour.



(c) VRSU births for 2000 vehicles per hour.

Fig. 62: The number of VRSU births over three hours for traffic driving at 55mph.

the bandwidth.

The first set of results are for vehicles in urban traffic that have slower velocities. Given vehicles are driving slower, the time spent in the VRSU service area will be longer, thus the results are expected to be better than that of the faster traffic in the highway traffic test.

Three values for flow are also considered, 500, 1000, and 2000 vehicles per hour, and are plotted separately in Figures 61(a), 61(b), and 61(c), respectively.

In the runs, each of the VRSU migrations will occur when the host vehicles is 500 meters from the end of the service area. As mentioned previously, if the value of $d_m \cdot 1.15$ is greater

than 500, then the migration will occur at this distance from the boundary of the service area. For the first set, this includes the following:

- Data Image Size: 30 MB, Bandwidth: 6Mbps, Migration Distance 690m,
- Data Image Size: 40 MB, Bandwidth: 6Mbps, Migration Distance 920m, and
- Data Image Size: 40 MB, Bandwidth: 10Mbps, Migration Distance 552m,

To our surprise, most of the runs resulted in a single VRSU, meaning during the entire 3 hour simulation, the VRSU stayed within the service area. To report the results, for each bandwidth and flow, we specify which is the maximum Data Image Size where there are 3 or less VRSU Births; this is equivalent to 1 VRSU Birth per hour of simulation.

For 27Mbps, 20Mbps, and 10Mbps each of the runs resulted in less than 3 VRSU births. In fact, all but the 40MB Data Image at 10Mbps resulted in only a single VRSU! For 6Mbps, all the runs resulted in less than 3 VRSU Births except for the 40 MB Data Image with a flow of 500 vehicles per hour.

From the first set of results, we expect the VRSU architecture to be able to reliably provide an aggregation service for urban traffic running the MO3-Flow method.

The second set of results are for vehicles in highway traffic that have faster velocities. Given these faster velocities, the time spent in the VRSU service area is expected to be longer, and the results will not be as good as the urban traffic test.

Three values for flow are also considered, 500, 1000, and 2000 vehicles per hour, and are plotted separately in Figures 62(a), 62(b), and 62(c), respectively.

In the runs, each of the VRSU migrations will occur when the host vehicles is 500 meters from the end of the service area. As mentioned previously, if the value of $d_m \cdot 1.15$ is greater than 500, then the migration will occur at this distance from the boundary of the service area. For the second set, this includes the following:

1. Data Image Size: 20 MB, Bandwidth: 6Mbps, Migration Distance 766.67m,
2. Data Image Size: 30 MB, Bandwidth: 6Mbps, Migration Distance 1150m,
3. Data Image Size: 30 MB, Bandwidth: 10Mbps, Migration Distance 690m,
4. Data Image Size: 40 MB, Bandwidth: 6Mbps, Migration Distance 1466.67m, and
5. Data Image Size: 40 MB, Bandwidth: 10Mbps, Migration Distance 920m,

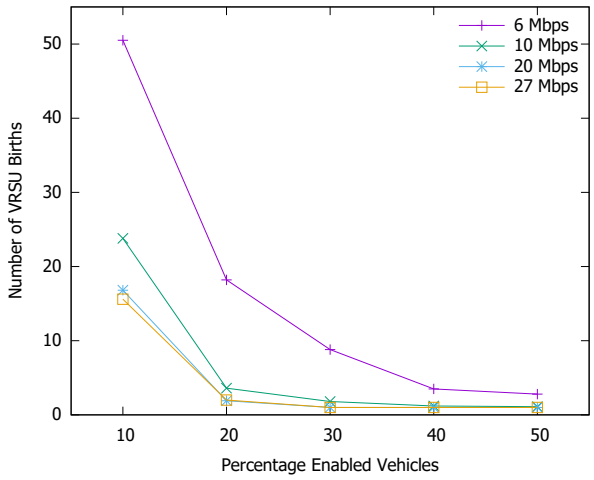


Fig. 63: Percentage enabled vehicles sensitivity analysis

Note that for items 2, 4, and 5, there are some migrations that are already required to migrate as soon as the VRSU becomes active. This results in a large number of VRSU births. To report the results, for each bandwidth and flow, we specify which is the maximum Data Image Size where there are 3 or less VRSU Births which is equivalent to 1 VRSU Birth per hour of simulation.

For bandwidths of 27Mbps and 20Mbps, the results were always under 3 VRSU Births. For 10Mbps, flows of 2000 and 1000 vehicles per hour had 3 VRSU Births with a Data Image of 30 MB and below. The flow of 500 vehicles per hour had 3 VRSU Births with a Data Image of 20 MB and below. For 5Mbps, flows of 2000 and 1000 vehicles per hour had 3 VRSU Births with a Data Image of 20MB and below. The flow of 500 vehicles per hour only had 3 VRSU Births with a Data Image of 10 MB.

The results from the second set show that the VRSU architecture is able to reliably provide an aggregation service for highway traffic running the MO3-Flow method.

7.3.3 SENSITIVITY ANALYSES

This section describes our sensitivity analysis of the percentage of enabled vehicles. It is important to know how the percentage of enabled vehicles affects the Number of VRSU Births. The results confirm what is expected, that with fewer enabled vehicles, the Number of VRSU Births increases.

In the previous results, the percentage of enabled vehicles was kept constant at 0.5. In this section we vary it between 0.1 and 0.5 by increments of 0.1. The Data Image size is set to 20 MB, the vehicle velocity is set to 55 mph (about 88.5 km per hour), and the flow of vehicles is set to 1000 vehicles per hour. Again, the simulation is run 10 times for each configuration and the average of the results are plotted in Figure 63.

For 10% enabled vehicles, all of the Number of VRSU Births is above 3. It starts at 50.5 with 5Mbps and drops down to 15.6 at 27Mbps. For 20% enabled vehicles, with 6 Mbps, there were 18.2 VRSU births and with 27Mbps, this drops down to 2 VRSU Births. For 30% enabled vehicles, with 6 Mbps, there are 8.8 VRSU Births and a 20Mbps, this falls to 1. For 40% enabled vehicles and 6Mbps, there are 3.5 VRSU Births and at 20Mbps, there is only 1 VRSU Birth. For 50% enabled vehicles, all of the Number of VRSU Births is below 3. It starts with 2.8 VRSU Births at 6Mbps and reaches 1 VRSU Birth at 20Mbps.

7.4 SUMMARY

The MO1, MO2, and MO3 methods each are able to determine traffic parameters without the assistance of infrastructure; however, the MO3-Flow method is different in that it does require data aggregation at the RSU. The contribution of this chapter is to introduce the VRSU architecture which was designed as a means to aggregate MO3-Flow data without requiring a physical RSU near the MO test area.

By minimizing the amount of data to be migrated to only the Data Image and migrating the Data Image to a new host in either the codirectional or oncoming traffic, the VRSU is able to remain in a service area reliably, despite the unreliable nature of the vehicular traffic.

Through simulation using simulated urban and highway traffic, we have shown that the VRSU architecture supports the migration of Data Images required for MO3-Flow and can in fact support larger services as well. This paves the way to additional services to be built using the VRSU architecture.

CHAPTER 8

CONCLUSION

Traffic parameter estimation is important for the TMC given its role in dynamic route guidance, incident detection, short-term travel prediction, and other Measures of Effectiveness parameters. Two classes of strategies have been developed to solve this problem. The first class is the stationary observer and the second is the moving observer, which is the strategy used throughout this work.

With the number of V2X compliant vehicles expected to increase in the upcoming years, and the increased sophistication of vehicles, USDOT has invested more and more in Connected Vehicle technology. This push and our own rediscovery of the moving observer method led to our main research question: Can a vehicle accurately estimate traffic parameters using onboard resources shared through CV technology in a lightweight manner without utilizing centralized or roadside infrastructure?

The Mobile Observer 1 Method is a modernized form of Wardrop and Charlesworth's Moving Observer method. In the MO1 method the concept of the tally is introduced to count codirectional traffic. The tally is defined as the number of times the test vehicle is passed by other vehicles minus the number of times the test vehicle passes other vehicles. The tallies and time to travel within a MO test area are shared between enabled vehicles and are aggregated to estimate average velocity, flow, and density. It turns out the simple concept of the tally can be used in other ways and is the base concept used in the proceeding methods.

In the Mobile Observer 2 Method, the concept of betweeness is introduced. In the method, an MO partnership is formed from two vehicles traveling in the same direction. If the number of vehicles between the two partners is known at some time t, then by sharing tallies, the number of vehicles between them at a later, or prior, time can be determined. We offer the observation that when two vehicles pass, the number of vehicles between them is typically, but not always zero. Using these tallies and the distance between the vehicles the density of vehicles between the two MO partners can be determined. Since vehicles are traveling in the same direction, the relative velocity between the two is low; it is found that typically longer distances are required to get good results.

In the Mobile Observer 3 Method, we started with a method similar to Mobile Observer 2; however, in this method the MO partners are traveling in opposite directions. By using

vehicles traveling in opposite directions, the relative velocity between the MO partners will be greater, and the method can be used in urban environments. In the method the MO partners will keep two tallies, one for codirectional traffic and another for oncoming traffic, which counts the number of oncoming vehicles it meets. In the method two fatal assumptions were made. The first is that the two MO partners will enter the test area at the same time. The second is that the results can simply be averaged over a time span to estimate the traffic. This resulted in poor results.

An alternative Mobile Observer 3 Flow method is then introduced where instead of measuring the density of the test area, the flow is measured. This removes the need for the assumption that vehicles enter the test area at the same time. Then a new way to aggregate results is introduced. The MO3-Flow method provides good results; however, it introduces a reliance on roadway infrastructure, ie. RSUs, to aggregate the results to estimate the flow.

The VRSU architecture is introduced to provide a solution to this issue. In the VRSU architecture, a host vehicle will run a service, ie. the Traffic Parameter Collection service, in a virtualized container. Then, as the host vehicle nears the boundary of the service area, it will migrate the service to another suitable host. By assuming the operating system and Service Image is preinstalled on the vehicle, the data to migrate is reduced to only the Data Image. This reduces the amount of time it takes to migrate and ensures the migration can be done successfully, thus making the service reliable.

8.1 CONTRIBUTIONS

The key contributions of this work are as follows:

1. A family of distributed algorithms, the Mobile Observer methods, make use of vehicles onboard computing, networking, and sensor resources to provide estimates of traffic parameters

    - The MO1 method provides estimates for flow, density, and average velocity of vehicles, assuming a constant flow of vehicles,
    - The MO2 method provides estimates of density on long stretches of roadways such as highways,
    - The MO3-Flow method provides estimates of flow for short stretches of urban roadways such as that between two traffic lights.

2. A VRSU architecture that can be used to provide supplemental coverage for RSUs that

can be provisioned quickly and does not require the cost associated with additional infrastructure.

Finally, we sum up these contributions and provide an answer to the main research question: Can a vehicle accurately estimate traffic parameters using onboard resources shared through CV technology in a lightweight manner without utilizing centralized or roadside infrastructure?

The MO1 method utilizes only V2V communications to share and aggregate data and showed good results for all three traffic parameters: flow, density, and velocity, even at low penetration rates.

The MO2 method also utilizes only V2V communication to share and aggregate data to estimate density. Again, the results are good even at low penetration rates.

The MO3 method does require an RSU to collect and aggregate data to estimate flow; however, with the introduction of the VRSU architecture, there is no need for roadside infrastructure, other than to coordinate the location of the VRSUs.

The research presented in this work has shown that the traffic parameters can in fact be accurately measured using vehicle sensor data without relying on centralized or roadside infrastructure; however, there are certain limitations in what has been presented. We now look at these limitations and conclude with additional future work.

8.2 LIMITATIONS AND FUTURE WORK

In spite of the simplicity of the methods and of the fact that it is, essentially V2V-based, the methods have a number of limitations that we now discuss.

The first limitation affects each of the MO methods. The limitation stems from the fact that our method is based on counting vehicle passes. It is clear that occluded vehicles cannot be counted. For MO1 and MO2, occlusion is inconsequential on two lane roads, and on three lane roads can be mitigated by only including tallies from vehicles in the center lane. However, it may easily become a problem on multi-lane roadways, especially at low penetration rate of enabled vehicles. For MO3 and MO3-Flow, the same issues exist for codirectional traffic, plus an additional issue that there may not be a direct line of sight of oncoming vehicles.

We assume that by networking together, enabled vehicles can keep each other informed of occlusions and can mitigate their effect. However, occlusions are likely to remain a challenge. In future work, we plan to evaluate the sensitivity of our method to counting errors induced by occlusions and to provide techniques for mitigating occlusion.

A second limitation of each of the MO methods is anchored in the conservation of flow requirement. If the conservation of flow is not guaranteed and vehicles are allowed to enter and leave the roadway at random, then proper tallies cannot be maintained and our method does not produce high quality estimates. A corollary of this is that in urban environments (say, on arterial corridors) our method is not guaranteed to work, with the exception of blocks where the conservation of flow can be enforced.

A third limitation of the methods is that the estimates are derived exclusively from passings. In very light traffic, the number of passings may be, generally, quite limited. Note that this affects MO1 and MO2 more than it does the MO3 methods. In MO3, the vehicle traveling in oncoming traffic is able to count each of those it meets. As a result, the methods do not yield accurate estimates in light traffic especially at low penetration rates. On the other hand, when traffic is light, obtaining a very precise traffic density or flow may not be as important as in the case of a well-traveled road.

In the future, we plan to use more sophisticated data aggregation strategies. In MO1, MO2, and MO3, the results are aggregated and averaged over a certain timespan. If the traffic flow varies, then with high traffic flow there is expected to be more enabled vehicles available and more results than when there is low traffic flow. When aggregating within a time span, there will be more results from the higher traffic, skewing the final result. In the future, work must be done to determine a better way to aggregate the results to mitigate this skew.

In addition to addressing these limitations, we now list some additional future work that can be addressed to enhance the MO methods and the VRSU architecture.

The MO3 method is shown to be adaptable for both density and flow. In future work, we will show that the MO2 method can also be adapted for both density and flow.

The MO3-Flow method aggregates the tallies or counts of vehicles passing into the test area. It does this using a two phase approach. In the first phase, a vehicle enters the test area at time t_1 and starts a tally. Then when passing an oncoming enabled vehicle, it will send its tally to the second vehicle to start phase 2. The first vehicle can keep counting tallies until it reaches the end of the test area at time t_4. Then an estimate of the tallies of vehicles entering the test area in the time span $[t_1, t_4]$ can be provided by the RSU or VRSU. With these two tallies, the density of the test area may be determined.

The VRSU was introduced in the context of the MO3-Flow Method. Here, it is ideal for the VRSU to remain static in location. However, keeping the VRSU static means the VRSU must be migrated frequently. In some circumstances, for example clustered highway traffic,

it may be beneficial for the VRSU to move with traffic. I relate the this to geosynchronous vs low-orbit satellites. Geosynchronous satellites remain stationary with respect to the Earth, whereas low-orbit satellites are always moving with respect to the Earth. This concept will be explored more in future work.

The concept of deploying VRSUs with the ability to communicate with one another using a message routing protocol like OPERA. Doing so would provide a reliable virtual infrastructure built over unreliable vehicular traffic.

The Mobile Observer Methods discussed in this work were born from the realization that much like vehicular clouds utilize the unused processing power of vehicles' CPUs, we can also utilize the unused data from the vehicles' sensors. In this work we focused only on sensors for counting passes. In fact, the VRSU results have shown that the Data Image of the VRSU for MO3-Flow is quite small in regards to the larger Data Image sizes tested. This means there is a capacity for services with larger data images or perhaps more such mini services utilizing data from some of the many other vehicles sensors that will provide even more benefit to the community. This will be an attractive field for continued research.

REFERENCES

[1] US Department of Transportation, "Pocket guide to transportation." https://www.bts.gov/pocketguide, 2021.

[2] American Society of Civil Engineers, "2017 Infrastructure Report Card - Roads." https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Roads-Final.pdf, 2017.

[3] NHTSA National Highway Traffic Safety Administration, "Traffic safety facts - preliminary 2009 report." http://www-nrd.nhtsa.dot.gov/Pubs/811255.pdf, March 2010.

[4] US Department of Transportation, Research and Innovative Technology Association, "National transportation statistics." http://www.bts.gov/publications/national_transportation_statistics/, 2011.

[5] Texas Transportation Institute, "Urban Mobility Report." http://mobility.tamu.edu/ums/, December 2019.

[6] US Federal Highway Administration, "Congestion pricing: A primer." http://www.ops.fhwaa.dot.gov/publications/congestionpricing/congestionpricing.pdf, 2006.

[7] R. Florin and S. Olariu, "A survey of vehicular communications for traffic signal optimization," *Vehicular Communications*, vol. 2, no. 2, pp. 70–79, 2015.

[8] 511 Traffic, "Latest traffic news." http://traffic.511.org/LatestNews.

[9] "Waze." http://www.waze.com, March 2017.

[10] S. M. Khan, K. C. Dey, and M. Chowdhury, "Real-time traffic state estimations with connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1687–1699, July 2017.

[11] J. Argote-Cabanero, E. Christofa, and A. Skabardonis, "Connected vehicle penetration rate for estimation of arterial measures of effectiveness," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 298–312, 2015.

[12] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe vehicle data for real-time traffic state estimation and short-term travel time prediction on a freeway," *Transportation Research Record*, vol. 1855, no. 1, pp. 49–59, 2003.

[13] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: a general approach," *Transportation Research Part B: Methodological*, vol. 39, no. 2, pp. 141–167, 2005.

[14] P. B. van Erp, V. L. Knoop, and S. P. Hoogendoorn, "Macroscopic traffic state estimation using relative flows from stationary and moving observers," *Transportation Research Part B: Methodological*, vol. 114, pp. 281 – 299, 2018.

[15] Y. Wang, M. Papageorgiou, and A. Messmer, "Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing," *Transportation Research Part A: Policy and Practice*, vol. 42, no. 10, pp. 1340–1358, 2008.

[16] T. Seo, A. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual Reviews in Control*, vol. 43, pp. 128–151, 2017.

[17] Federal Highway Administration, US Department of Transportation, "Highway Performance Monitoring System." http://www.fhwa.dot.gov/policyinformation/hpms/volumeroutes/ch4.cfm, September 2014.

[18] J. de Dios Ortúzar and L. G. Willumsen, *Modelling Transport*. Wiley, 2002.

[19] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering*. Pearson Prentice Hall, 4-th Edition, 2011.

[20] I. Sreedevi and J. Black, "Loop detectors." California Center for Innovative Transportation, feb 2001.

[21] M. Fontaine, "Traffic monitoring," in *Vehicular Networks: From Theory to Practice* (S. Olariu and M. C. Weigle, eds.), ch. 1, pp. 1.1 – 1.28, Boca Raton, Florida: Taylor and Francis, 2009.

[22] R. Sengupta, S. Rezaei, S. E. Shlavoder, D. Cody, S. Dickey, and H. Krishnan, "Cooperative collision warning systems: Concept definition and experimental implementation." California PATH Technical Report UCB-ITS-PRR-2006-6, May 2006.

[23] Z.-X. Li, X.-M. Yang, and Z. Li, "Application of cement-based piezoelectric sensors for monitoring traffic flows," *Journal of Transportation Engineering*, vol. 132, no. 7, pp. 565–573, 2006.

[24] G. Yan, S. Olariu, and D. C. Popescu, "NOTICE: An Architecture for the Notification of Traffic Incidents," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, pp. 6–16, Winter 2012.

[25] NHTSA National Highway Traffic Safety Administration, "An analysis of recent improvements to vehicle safety." http://www-nrd.nhtsa.dot.gov/Pubs/811572.pdf, June 2012.

[26] J. G. Wardrop and G. Charlesworth, "A method of estimating speed and flow of traffic from a moving vehicle," *Proceedings of the Institution of Civil Engineers, Part II*, vol. 3, pp. 158–171, 1954.

[27] A. Mulligan and A. Nicholson, "Uncertainty in traffic flow estimation using the moving observer method," in *Institution of Professional Engineers New Zealand (IPENZ) Transportation Group. Technical Conference Papers 2002*, (Rotorua, New Zealand), 2002.

[28] C. Wright, "A theoretical analysis of the moving observer method," *Transportation Research*, vol. 7, pp. 293–311, 1973.

[29] C. A. O'Flaherty and F. Simons, "An evaluation of the moving observer method of measuring traffic speeds and flow," in *Proc. 5-th Conference of the Australian Road Research Board*, pp. 40–54, 1970.

[30] US Department of Transportation, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications." ASTM E2213-03, September 2003.

[31] K. Arai and S. R. Sentinuwo, "Method for traffic flow estimation using on-dashboard camera image," *Computer Science and Aplications*, vol. 3, no. 2, pp. 18–22, 2014.

[32] W. C. Chang and K. J. Hsu, "Vision-based side vehicle detection from a moving vehicle," in *System Science and Engineering (ICSSE), 2010 International Conference on*, pp. 553–558, July 2010.

[33] R. Stevenson, "A driver's sixth sense," *IEEE Spectrum*, vol. 348, pp. 50–55, October 2011.

[34] R. Florin and S. Olariu, "On a variant of the mobile observer method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 441–449, Feb 2017.

[35] R. Florin and S. Olariu, "Towards real-time density estimation using vehicle-to-vehicle communications," *Transportation Research Part B: Methodological*, vol. 138, pp. 435–456, Aug 2020.

[36] Tesla, "Auto pilot." https://www.Tesla.com/autopilot, 2021.

[37] T. S. Perry, "Will camera startup light give autonomous vehicles better vision than lidar?." https://spectrum.ieee.org/view-from-the-valley/transportation/sensors/will-camera-startup-light-give-autonomous-vehicles-better-vision-than-lidar, August 2021.

[38] Ford Motor Company, "Sustainability report summary 2013/14." http://corporate.ford.com/microsites/sustainability-report-2013-14/vehicle-avoidance.html, 2013.

[39] C. Wardlaw, "What is ford co-pilot 360?." https://www.jdpower.com/cars/shopping-guides/what-is-ford-co-pilot-360, October 2020.

[40] US Department of Transportation, "Connected vehicles." https://www.its.dot.gov/research_areas/connected_vehicle.htm, accessed December 2017.

[41] US Department of Transportation, "Connected vehicles basics." https://www.its.dot.gov/cv_basics/cv_basics_20qs.htm, 2018.

[42] Intelligent Transportation System Joint Program Office, "ITS Strategic Plan 2015-2019." http://www.its.dot.gov/factsheets/pdf/ITS_JPO_StratPlan.pdf, 2015.

[43] Intelligent Transportation System Joint Program Office, "Connected vehicles: Benefits, roles, outcomes." http://www.its.dot.gov/research_areas/WhitePaper_connected_vehicle.htm.

[44] US Department of Transportation, National Highway Traffic Safety Administration, "Vehicle-To-Vehicle Communication Technology." Technical Report No.11078-101414-v2a, 2014.

[45] Douglas Gettman, "DSRC and C-V2X: Similiarities, Differences, and the Future of Connected Vehicles." https://www.kimley-horn.com/news-insights/dsrc-cv2x-comparison-future-connected-vehicles/, June 2020.

[46] S. Zeadally and M. A. Javed, "Vehicular communications for ITS: Standardization and challenges," *IEEE Communications Standards Magazine*, vol. 4, 09 2019.

[47] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, pp. 1162–1182, July 2011.

[48] SAE International, "Dedicated short range communications (DSRC) message set dictionary." SAE Standard J2735_201601, Jan 2016.

[49] "U.S. Federal Communications Commission MO&O, amendment of the commission's rules regarding dedicated short-range communication services in the 5.850–5.925 GH band (5.9 GHz band)." FCC 06-110, adopted Jul. 20, 2006.

[50] 3rd Generation Partnership Project, "Release 14." https://www.3gpp.org/release-14, March 2016.

[51] Federal Communications Commission, "In the matter use of the 5.850-5.925 GHz band." https://docs.fcc.gov/public/attachments/FCC-20-164A1.pdf, November 2020.

[52] J. Kurose and K. Ross, *Computer Networking: A top down approach, 6th Edition.* Boston: Pearson, 2013.

[53] P. Ghazizadeh, R. Florin, A. G. Zadeh, and S. Olariu, "Reasoning about mean time to failure in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 751–761, 2016.

[54] R. Florin, P. Ghazizadeh, A. Ghazi Zadeh, S. El-Tawab, and S. Olariu, "Reasoning about job completion time in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1762–1771, 2017.

[55] R. Florin, A. Ghazi Zadeh, P. Ghazizadeh, and S. Olariu, "Towards approximating the mean time to failure in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2045–2054, 2018.

[56] R. Florin, A. Ghazizadeh, P. Ghazizadeh, S. Olariu, and D. C. Marinescu, "Enhancing reliability and availability through redundancy in vehicular clouds," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[57] R. Florin and S. Olariu, "Toward approximating job completion time in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 3168–3177, 2019.

[58] R. Florin, P. Ghazizadeh, A. G. Zadeh, R. Mukkamala, and S. Olariu, "A tight estimate of job completion time in vehicular clouds," *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, pp. 721–734, 2020.

[59] R. Florin and S. Olariu, "Vehicular clouds: A view from above," in *Vehicular Cloud Computing for Traffic Management and Systems*, pp. 1–29, 2018.

[60] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing*. Boston: Morgan Kaufmann, 2013.

[61] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1587–1595, May 2014.

[62] P. Ghazizadeh, "Resource allocation in vehicular cloud computing." PhD Thesis, Old Dominion University, July 2014.

[63] T. K. Refaat, B. Kantarci, and H. T. Mouftah, "Virtual machine migration and management for vehicular clouds," *Vehicular Communications*, vol. 4, pp. 47–56, April 2016.

[64] B. Baron, M. Campista, P. Spathis, L. H. Costa, M. Dias de Amonim, O. C. Duarte, G. Pujolle, and Y. Viniotis, "Virtualizing vehicular node resources: Feasibility study of virtual machine migration," *Vehicular Communications*, vol. 4, pp. 39–46, April 2016.

[65] Linux Containers, "What's LXC." https://linuxcontainers.org/lxc/introduction/, 2021.

[66] S. Wang, J. Xu, N. Zhang, and Y. Liu, "A survey on service migration in mobile edge computing," *IEEE Access*, vol. 6, 2018.

[67] R. Morabito, R. Petrolo, V. Loscrì, N. Mitton, G. Ruggeri, and A. Molinaro, "Lightweight virtualization as enabling technology for future smart cars," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 1238–1245, 2017.

[68] G. F. Newell, "A simplified theory of kinematic waves in highway traffic, part I: General theory," *Transportation Research Part B: Methodological*, vol. 27, no. 4, pp. 281 – 287, 1993.

[69] Y. Makigami, G. F. Newell, and R. Rothery, "Three-dimensional representation of traffic flow," *Transportation Science*, vol. 5, no. 3, pp. 302–313, 1971.

[70] T. Seo and T. Kusakabe, "Probe vehicle-based traffic state estimation method with spacing information and conservation law," *Transportation Research Part C: Emerging Technologies*, vol. 59, pp. 391 – 403, 2015.

[71] E. F. Grumert and A. Tapani, "Traffic state estimation using connected vehicles and stationary detectors," *Hindawi Journal of Advanced Transportation*, pp. 1–14, 2018.

[72] US Department of Transportation, "SAE J2735 Standard: Applying the Systems Engineering Process." https://www.its.dot.gov/index.htm, January 2013.

[73] J. C. Herrera, D. B. Work, R. Herring, X. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: the mobile century field experiment," *Transportation Research, Part C: Emerging Technologies*, vol. 18, no. 4, pp. 566–583, 2010.

[74] S. Tao, V. Manolopoulos, S. Rodriguez, and A. Rusu, "Real-time urban traffic state estimation with A-GPS mobile phones as probes," *Journal of Transportation Technologies*, vol. 2, pp. 22–31, 2012.

[75] N. Uno, F. Kurauchi, and H. Tamura, "Using bus probe data for analysis of travel time variability," *Journal of Intelligent Transportation Systems*, vol. 13, no. 1, pp. 2–15, 2009.

[76] S. Sunkari, "The benefits of retiming traffic signals," *ITE Journal*, pp. 26–29, 2004.

[77] US Department of Transportation, Federal Highway Administration, "Corridor simulator (CORSIM/TSIS)." http://ops.fhwa.dot.gov/trafficanalysistools/corsim.htm, 2006.

[78] A. Stevanovic, "Adaptive traffic control systems: Domestic and foreign state of practice," Tech. Rep. Synthesis 403, National Cooperative Highway Research Program, 2010.

[79] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using dynamic Bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1679–1693, December 2012.

[80] J. de Dios Ortúzar and L. G. Willumsen, *Modelling Transport*. New York: Wiley, 2002.

[81] P. Koonce, L. Rodegerdts, K. Lee, and S. Quayle, *Traffic Signal Timing Handbook, Final Report, FHWA Contract No. DTFH61-98-C-00075*. Portland OR: Kittelson and Associates Inc, June 2008.

[82] K. Kwong, R. Kavaler, R. Rojagopal, and P. Varaiya, "A practical scheme for arterial travel time estimation based on vehicle re-identification using wireless sensors," *Transportation Research Board, Washington D.C.*, 2009.

[83] D. Kari, W. Guoyuan, and M. Barth, "Eco-friendly freight signal priority using connected vehicle technology: A multi-agent systems approach," *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 1187–1192, 2014.

[84] P. Bhuvaneswari, G. Raj, R. Balaji, and S. Kanagasabai, "Adaptive traffic signal flow control using wireless sensor networks," *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*, pp. 85–89, 2012.

[85] C. Wenjie, C. Lifeng, C. Zhanglong, and T. Shiliang, "A realtime dynamic traffic control system based on wireless sensor network," *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*, pp. 258–264, 2005.

[86] S. Kwatirayo, J. Almhana, and Z. Liu, "Optimizing intersection traffic flow using vanet," *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pp. 260–262, 2013.

[87] N. Goodall, *Traffic Signal Control with Connected Vehicles*. PhD thesis, Department of Civil Engineering, University of Virginia, 2013.

[88] M. Khamis and W. Gomaa, "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 134–151, 2014.

[89] D. Prasad, "Adaptive traffic signal control system with cloud computing based online learning," *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pp. 1–5, 2011.

[90] D. McKenney and T. White, "Distributed and adaptive traffic signal control within a realistic traffic simulation," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 574–583, 2013.

[91] S. Cheng, M. Epelman, and R. Smith, "Cosign: A parallel algorithm for coordinated traffic signal control," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 4, pp. 551–564, 2006.

[92] M. K. Abbas, M. N. Karsiti, M. Napiah, B. B. Samir, and M. Al-Jemeli, "High accuracy traffic light controller for increasing the given green time utilization," *Computers & Electrical Engineering*, vol. 41, pp. 40–51, 2015.

[93] "Connected vehicle research." http://www.its.dot.gov/connected_vehicle/connected_vehicle.htm, October 2014.

[94] N. Maslekar, J. Mouzna, M. Boussedjra, and H. Labiod, "Cats: An adaptive traffic signal system based on car-to-car communication," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1308–1315, 2013.

[95] B. Xiang-yu, Y. Xin-min, J. Hai, and L. Jun, "A novel traffic information system for vanet based on location service," *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, pp. 1–6, 2008.

[96] G. Comert and M. Cetin, "Queue length estimation from probe vehicle location and the impacts of sample size," *European Journal of Operational Research*, vol. 197, no. 1, pp. 196–202, 2009.

[97] G. Comert, "Simple analytical models for estimating the queue lengths from probe vehicles at traffic lights," *Transportation Research Part B: Methodological*, vol. 55, pp. 59–74, 2013.

[98] J. Q. Li, K. Zhou, S. E. Shladover, and A. Skabardonis, "Estimating queue length under connected vehicle technology: Using probe vehicle, loop detector, and fused data," *Transportation Research Record*, vol. 2356, no. 1, pp. 17–22, 2013.

[99] Y. Cheng, X. Qin, J. Jin, B. Ran, and J. Anderson, "Cycle-by-cycle queue length estimation for signalized intersections using sampled trajectory data," *Transportation Research Record*, vol. 2257, no. 1, pp. 87–94, 2011.

[100] S. E. Shladover and J.-Q. Li, "Evaluation of probe vehicle sampling strategies for traffic signal control," in *Proc. 14th IEEE ITSC Conference*, (Washington, DC), October 2011.

[101] Q. Cai, Z. Wang, L. Zheng, B. Wu, and Y. Wang, "Shock wave approach for estimating queue length at signalized intersections by fusing data from point and mobie sensors," *Transportation Research Record*, vol. 2422, no. 1, pp. 79–87, 2014.

[102] P. Hao and X. Ban, "Vehicle queue location estimation for signalized intersections using sample travel times from mobile sensors." 2011 Transportation Research Board 90-th Annual Meeting, December 2011.

[103] J. Zheng and H. X. Liu, "Estimating traffic volumes for signalized intersections using connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 347 – 362, 2017.

[104] V. V. Gayah and V. V. Dixit, "Using mobile probe data and the macroscopic fundamental diagram to estimate network densities: Tests using microsimulation," *Transportation Research Record*, vol. 2390, no. 1, pp. 76–86, 2013.

[105] T. Seo, Y. Kawasaki, T. Kusakabe, and Y. Asakura, "Fundamental diagram estimation by using trajectories of probe vehicles," *Transportation Research Part B: Methodological*, vol. 122, pp. 40 – 56, 2019.

[106] G. Fusco, C. Colombaroni, and N. Isaenko, "Short-term speed predictions exploiting big data on large urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 183 – 201, 2016.

[107] S. H. Lim, Y. K. Chia, and L. Wynter, "Accurate and cost-effective traffic information acquisition using adaptive sampling: Centralized and V2V schemes," *Transportation Research Part C: Emerging Technologies*, vol. 94, pp. 99 – 120, 2018.

[108] C. Wang, B. Ran, H. yang, J. Zhang, and X. Qu, "A novel approach to estimate freeway traffic state: Parallel computing and improved Kalman fiter," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, pp. 180–193, April 2018.

[109] T. Qiu, X.-Y. Lu, A. Chow, and S. E. Shladover, "Estimation of freeway traffic density with loop detector and probe vehicles," *Transportation Research Record: Journal of Transportation Research Board*, pp. 21–29, 2010.

[110] H. Van Lint and S. Hoogendoorn, "A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, no. 8, pp. 596–612, 2010.

[111] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1156 – 1166, September 2012.

[112] A. Anand, G. Ramadurai, and L. Vanajakshi, "Data fusion-based traffic density estimation and prediction," *Journal of Intelligent Transportation Systems*, vol. 18, no. 4, pp. 367–378, 2014.

[113] J. Zhang, S. He, W. Wang, and F. Zhan, "Accuracy analysis of freeway traffic speed estimation based on the integration of cellular probe system and loop detectors," *Journal of Intelligent Transportation Systems*, vol. 19, no. 4, pp. 411–426, 2015.

[114] L. Ambuehl and M. Menendez, "Data fusion algorithm for macroscopic fundamental diagram estimation," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 184 – 197, 2016.

[115] M. Fountoulakis, N. Bekiaris-Liberis, C. Roncoli, I. Papamichail, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing," *Transportation Research Part C: Emerging Technologies*, vol. 78, pp. 13 – 33, 2017.

[116] A. M. Bayen, J. Butler, and A. D. Patire, "Mobile Millenium Final Report," Tech. Rep. UCB-ITS-CWP-2011-6, California Center for Innovative Transportation, September 2011.

[117] M. C. Weigle and S. Olariu, "Intelligent highway infrastructure for planned evacuations," in *Proceedings of the IEEE International Workshop on Research Challenges in*

Next Generation Networks for First Responders and Critical Infrastructures (NetCri), (New Orleans, LA), April 2007.

[118] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: From reservation- to optimization-based methods," *Transportation Research Part B: Methodological*, vol. 122, pp. 416–435, 2019.

[119] N. Bekiaris-Liberis, C. Roncoli, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3484–3497, 2016.

[120] G. Yan and S. Olariu, "A probabilistic analysis of link duration in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 1–10, 2011.

[121] F. Zheng, S. E. Jabari, H. X. Liu, and D.-C. Lin, "Traffic state estimation using stochastic Lagrangian dynamics," *Transportation Research Part B: Methodological*, vol. 115, pp. 143 – 165, 2018.

[122] P. B. C. van Erp, S. Thoen, V. L. Knoop, and S. P. Hoogendoorn, "Estimating the fundamental diagram using moving observers," in *Proc. 21-st International Conference on Intelligent Transportation Systems (ITSC'2018)*, pp. 1974–1980, November 2018.

[123] K. Hong and J. Kenney and V. Rai and K. Laberteaux, "Evaluation of multi-channel schemes for vehicular safety communications," in *in Proc. 3rd IEEE Int. Symp. Wireless Veh. Commun. (WiVEC 2010)*, (Taipei), May 2010.

[124] US Department of Transportation, "NGSIM new generation simulation." http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, accessed August 2019.

[125] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.

[126] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125, 2018.

[127] M. Abuelela, S. Olariu, and I. Stojmenovic, "OPERA: Opportunistic packet relaying in disconnected vehicular ad hoc networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 285–294, 2008.

VITA

Ryan Florin
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

EDUCATION

PhD	Computer Science	Old Dominion Univeristy	August 2021
MS	Computer Science	Old Dominion Univeristy	May 2011
BS	Computer Science	Old Dominion Univeristy	May 2005

PROFESSIONAL EXPERIENCE

October 2015 - Current	Technical Project Manager	IQVIA	Norfolk, VA
August 2012 - October 2015	Software Engineer	IQVIA	Norfolk, VA
August 2011 - August 2012	Technical Lead	IQVIA	Norfolk, VA
August 2008 - August 2011	Software Developer	IQVIA	Norfolk, VA

PUBLICATIONS

See https://scholar.google.com/citations?user=HNicVaEAAAAJ

Typeset using LaTeX.