BSM (+BMM) Data Emulator

Dynamic Interrogative Data Capture (DIDC) Assessment Report: Impacts of DIDC

www.its.dot.gov/index.htm
Final Report (Ver 1.0) — April 2016
Publication Number: FHWA-JPO-17-516

U.S. Department of Transportation

Produced by Noblis, Inc.
U.S. Department of Transportation
ITS Joint Program Office
Research and Innovative Technology Administration

All tables, graphs, and artwork created by Noblis, Inc.

Note: All tables and figures are original to this report.

Notice

Technical Report Documentation Page

1. Report No. **FHWA-JPO-17-516**	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle **BSM (+BMM) Data Emulator Dynamic Interrogative Data Capture (DIDC) Assessment Report: Impacts of DIDC**		5. Report Date **April 2016** 6. Performing Organization Code
7. Author(s) **Taylor Deurbrouck, James O'Hara, and Karl Wunderlich**		8. Performing Organization Report No.
9. Performing Organization Name And Address Noblis 600 Maryland Ave., SW, Suite 755 Washington, DC 20024		10. Work Unit No. (TRAIS) 11. Contract or Grant No. DTFH61-11-D-00018
12. Sponsoring Agency Name and Address ITS-Joint Program Office Office of the Assistant Secretary for Research and Technology, USDOT 1200 New Jersey Avenue, S.E. Washington, DC 20590		13. Type of Report and Period Covered Final Report, Version 1 14. Sponsoring Agency Code **HOIT-1**
15. Supplementary Notes Work Performed for: Jim McCarthy, FHWA Resource Center		

16. Abstract

The objective of the Dynamic Interrogative Data Capture (DIDC) algorithms and software is to optimize the capture and transmission of vehicle-based data under a range of dynamically configurable messaging strategies. The key hypothesis of DIDC is that using a well-constructed DIDC approach will have a lower risk of privacy issues and higher effective measures estimation along with a reduced data transmission load than comparable non-DIDC alternatives. The DIDC method will go through two rounds of testing using a simulation network: the first round of experimentation will study DIDC concepts and the second round will compare DIDC to other message types.

The purpose of this report is to describe the testing method and results from comparing the DIDC messaging concept with current message types. This is the second phase of DIDC testing conducted in the BSM Data Emulator project. The first phase of DIDC testing, the proof of concept, was completed in January 2016 and is described in a separate test plan and final report. This second phase of testing compared the measures estimation capabilities, data communication load, and re-identification efforts of the DIDC concept to Basic Safety Message (BSM), Probe Data Message (PDM), and the European Union Cooperative Awareness Message (CAM).

17. Key Words Connected Vehicle, Basic Safety Message, Modeling and Simulation, Intelligent Transportation Systems, Evaluation, Vehicle Messaging, Probe Data Message, Dynamic Interrogative Data Capture, Basic Mobility Message, Cooperative Awareness Message	18. Distribution Statement		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 42	22. Price

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

Table of Contents

List of Tables

List of Figures

Executive Summary

DIDC algorithms and software is a key product of the Basic Safety Message (BSM) Data Emulator project. The BSM Data Emulator project is one of several related research and development activities within the Data Capture and Management (DCM) Program, which is in turn a part of the USDOT *connected vehicle* research effort considering mobile data communications in surface transportation to improve safety, mobility, and the environment. This assessment report pertains to a closely related application of DIDC product:

- TCA-DIDC: Offline software built with the purpose of interfacing with the Trajectory Conversion Algorithm (TCA) Version 2 Software Release 4 expected in April 2016.

The purpose of this report is to describe the scenarios, testing method, and results from comparing the DIDC messaging concept with other message types available in the TCA-DIDC Version 2.4 software. This is the second phase of DIDC testing conducted in the BSM Data Emulator project. The first phase of DIDC testing, the proof of concept, was completed in January 2016 and is described in a separate test plan and final report. This second phase of testing compared the measures estimation capabilities and data communication load of the DIDC concept to the European Union Cooperative Awareness Message (CAM) as well as the enhanced versions of the Basic Safety Message (BSM) and Probe Data Message (PDM) developed in the Measures Estimation effort of the BSM Data Emulator project.

Purpose

The purpose of this report is to describe the testing method and results from comparing the DIDC messaging concept with current message types. Elements that are covered include:

- Assumptions
- Research Hypotheses
- Technical Approach
- Analyses Scenarios and
- Results Reporting

Technical Approach

Three factors were considered for measuring the usefulness, efficiency, and privacy of each message type: key performance measurement estimation, data communication costs, and re-identification accuracy. The effectiveness of each message type was measured by its ability to provide vehicle data that produced accurate performance measure estimation of four key measures: travel time, queues, slippery conditions, and turning movements. Another important variable for each test scenario was the data communication cost calculated by how much vehicle data is transmitted during the simulation. And finally, Noblis produced a re-identification algorithm written in the python programming language to run on all of the various message types tested to determine if a vehicle could be tracked to its origin or destination. This algorithm was designed to accept TCA output of any of the four message types and predict a vehicle's origin and destination using the available message elements. For each analysis scenario, the re-identification algorithm attempted to track five separate vehicles starting with a vehicle message generated in the middle diamond area (see Figure ES- 1).

Analysis Scenarios

Noblis developed a test network in VISSIM called Philosopher's Corner to test the various message types. Figure ES- 1 shows the key features of the network which include eighteen origin and destination points spread out amongst two towns, Platoville and Spinoza Oaks, with a shopping center in the middle.

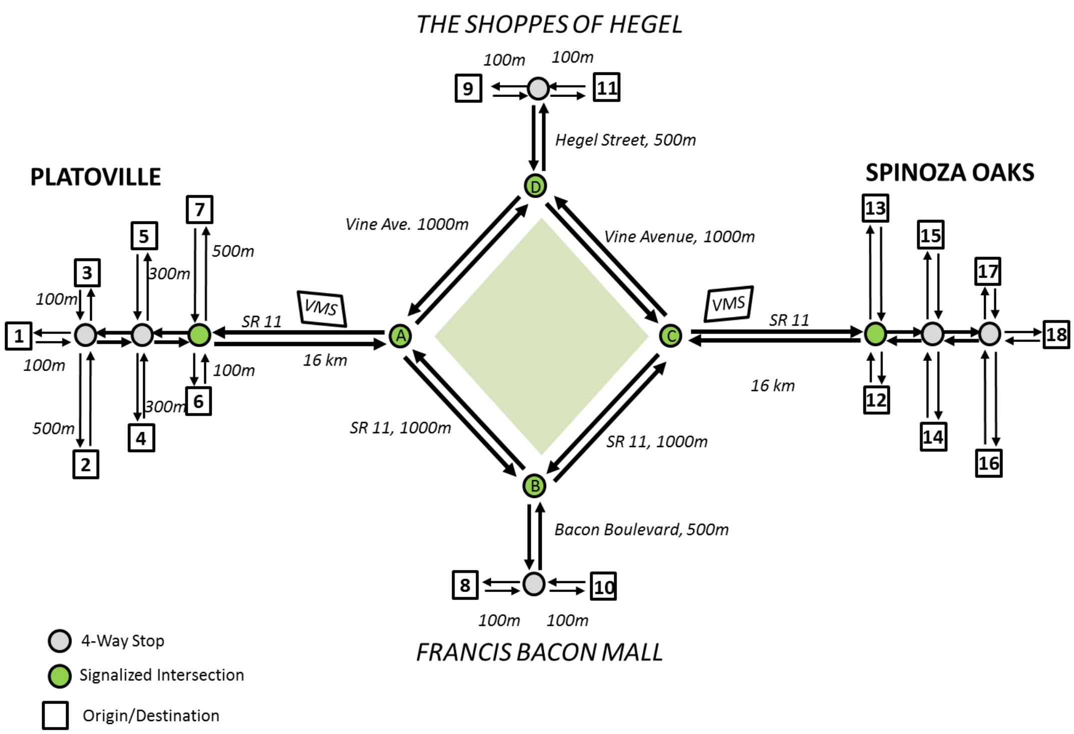

Figure ES- 1: Philosopher's Corner VISSIM Test Network

Key Findings

- **The DIDC concept generated data more efficiently than other message types**: The DIDC concept effectively throttled up data in low market penetration simulations and reduced redundant data in high market penetrations without sacrificing measures estimation capabilities for travel times, queues, turning movements, and slippery conditions.

- **The DIDC concept produced BMMs that were harder to re-identify the vehicle origin and destination compared to BSM and CAM, but easier than PDM:** A vehicle's origin and destination could be re-identified less than 35% of the time using BMMs at market penetrations greater than 20% on the Philosopher's Corner network. By comparison, the origin and destination could be re-identified more than 90% of the time using BSMs but less than 20% of the time using PDMs.

- **The TCA Version 2.4 software tool provides capability for others to build upon and find better ways of optimization:** The testing documented and completed in the BSM Emulator project looked at simple adaptive processes primarily to see if the measures estimation results were preserved or improved while reducing data flow. There is apparent potential value in these types of adaptive processes based on the research outlined in this report. Much more research could be explored in this area and the TCA Version 2.4 tool provides a good starting point for continuing investigation.

1 Introduction

Data from wirelessly connected vehicle and mobile devices have the potential to enable new applications and transform transportation systems management, traveler safety, and personal mobility. However, the utility of any new application is dependent on the underlying process by which mobile-source data are generated, stored, communicated, and mined for information. The USDOT's vision for connected vehicle systems leverages a regulatory DSRC-based Basic Safety Message (BSM) augmented with market-driven messages passed through alternative, longer-range communications media (e.g., cellular). Efforts within the Data Capture and Management (DCM) Program to evaluate alternative messaging protocols and innovations is a critical cross-cutting effort at the heart of establishing a new wirelessly connected vehicle/traveler paradigm. A wide range of critical research questions remain to be addressed, including the identification of required data elements, the roles of dual-mode communications and messaging protocols in detecting and or predicting traffic phenomena.

As part of the Basic Safety Message (BSM) Data Emulator project, Noblis researchers have developed the Dynamic Interrogative Data Capture (DIDC) concept. DIDC was first proposed by Noblis researchers as a possible method of reducing bandwidth costs by tailoring data communications adaptively to meet application needs. Noblis researchers added the DIDC capability to the Trajectory Conversion Algorithm (TCA) software and developed algorithms to estimate key transportation measures. These algorithms and software were used to test the DIDC concept under a variety of scenarios to refine the concept and prove its effectiveness in both accurately predicting measures of performance (maximizing the value of the data) while minimizing the amount of data captured and transmitted (reducing data-related costs).

Another key goal of the project is to conduct a side-by-side comparison of the measures estimation results and data communication load of DIDC with other message types including the European Union Cooperative Awareness Message (CAM) and the enhanced versions of the BSM and PDM introduced in the Measures Estimation effort of the BSM Data Emulator project. This assessment will inform the USDOT of the value of DIDC compared to current messaging strategies. The key hypothesis of DIDC is that using a well-constructed DIDC approach will have a lower risk of privacy issues and higher effective measures estimation along with a reduced data transmission load than comparable non-DIDC alternatives.

1.1 Purpose

The purpose of this report is to describe the testing method and results from comparing the DIDC messaging concept with current message types. Elements that are covered include:

- Assumptions
- Research Hypotheses
- Technical Approach
- Analyses Scenarios and
- Results Reporting

2 Assumptions

It is critical to document assumptions to caveat findings from the analyses so that there are no false expectations of the benefits that may be realized in the field. The assumptions of this study are the following:

1. PDM privacy protocols modeled in study will follow SAE J2735 standards [1]
    a. Probe Segment Number (PSN) changes every 120 seconds or 1 km, whichever comes later
    b. After PSN changes, no snapshots are generated for 3 to 13 seconds, or 50 to 250 meters, whichever comes first
2. DSRC communication is not represented. The enhanced versions of the BSM and PDM recommended by the Measures Estimation task include Dual Mode communication. That is, the vehicle transmits via DSRC if in range, else transmit via cellular. For simplicity's sake, only cellular communication will be used in these analyses since the transmission frequency will not be affected.
3. Cellular communication loss and latency are not represented.
4. Cost of cellular coverage is not part of the assessment.

3 Research Hypotheses

This section describes the key research hypotheses that will guide the development of the analysis scenarios. The hypotheses listed below are valid under assumptions, specifically assumption 4 that pertains to cost of deployment, listed in the previous section. For each hypothesis, other critical factors are held constant during targeted testing. For example, the slippery condition regions are held constant when exploring the effect of changing market penetration.

1. BMM will be able to produce comparable measures estimation results to BSM and CAM.
2. BMM will be able outperform PDM in measures estimation.
3. BMM will be able to produce a lower data load than BSM and CAM.
4. Vehicle origins and destinations will be easier to re-identify using BSMs than other message variants.
5. The DIDC Controller will be able to adjust BMM generation rates to match the defined targets.

4 Technical Approach

This section identifies the key transportation measures and variables that were considered and examined. Three factors were considered for measuring the usefulness, efficiency, and privacy of each message type: key performance measurement estimation, data communication costs, and re-identification accuracy. Each of these three variables are explained below.

4.1 Performance Measures

The effectiveness of each message type was measured by its ability to provide vehicle data that produced accurate performance measure estimation of four key transportation measures. These four key measures were chosen because they cover all of the significant aspects of the DIDC concept and were used in the first phase of DIDC testing. These measures and their descriptions are:

Queues

- A vehicle is in queue when it is either stopped or is traveling at a speed less than 10 ft/s (3 m/s) and is approaching another queued vehicle at headway of less than 20 ft (6 m).

Travel time

- This is defined as the average travel time on route segments experienced by all vehicles that begin travel in a specific time interval.

Slippery Conditions

- The use of traction control suggests a possible occurrence of slippery conditions. Slippery conditions are defined by the start and end time as well as the roadway coordinates of where they start and end.

Turning Movements

- This is defined as a percentage of vehicles turning left or right at a given intersection.

Key transportation measures were estimated by running the measures estimation algorithms on the simulated vehicle-based messages output from the TCA software. To ensure consistency in measures estimation analysis, Noblis researchers used algorithms to calculate the ground truth measures from the traffic simulation outputs that describe the vehicle dynamics (e.g., position, speed, acceleration rates) of every vehicle in the network.

Two of these algorithms, travel time and queue, were developed during the measures estimation task of the BSM Emulator project [4]. The remaining two algorithms, slippery conditions and turning movements, were developed during the first phase of DIDC testing to study the DIDC proof of concept. All four algorithms were written to work with all four message types.

4.2 Data Communication Cost

Another important variable for each test scenario was the data communication cost calculated by how much vehicle data is transmitted during the simulation. One of the assumptions in the Phase 2 test scenarios was that all communication was unlimited by loss, latency, or bandwidth. However, the amount of vehicle data was still a factor in ranking the efficiency of the message types against each other.

4.3 Re-Identification Accuracy

Re-identification is the capability of predicting a vehicle's origin and destination from a single data point anywhere on the network using all of the transmitted vehicle messages. Noblis produced a re-identification algorithm written in the python programming language to run on all of the various message types tested in Phase 2 to determine if a vehicle could be tracked to its origin or destination. This algorithm was designed to accept TCA output of any of the four message types and predict a vehicle's origin and destination using the available message elements. For each analysis scenario, the re-identification algorithm attempted to track five separate vehicles starting with a vehicle message generated in the middle diamond area (see Figure 5-1). These algorithms will be documented in Appendix B.

5 Analysis Scenarios

This section presents the analysis scenarios and the VISSIM test network used in the Phase 2 DIDC testing.

5.1 Test Network

Using the requirements for the DIDC test network, Noblis developed Philosopher's Corner as depicted in Figure 5-1. This VISSIM network was designed to meet the requirements outlined in Section 2. Key features include eighteen origin and destination points spread out amongst two towns, Platoville and Spinoza Oaks, with a shopping center in the middle.

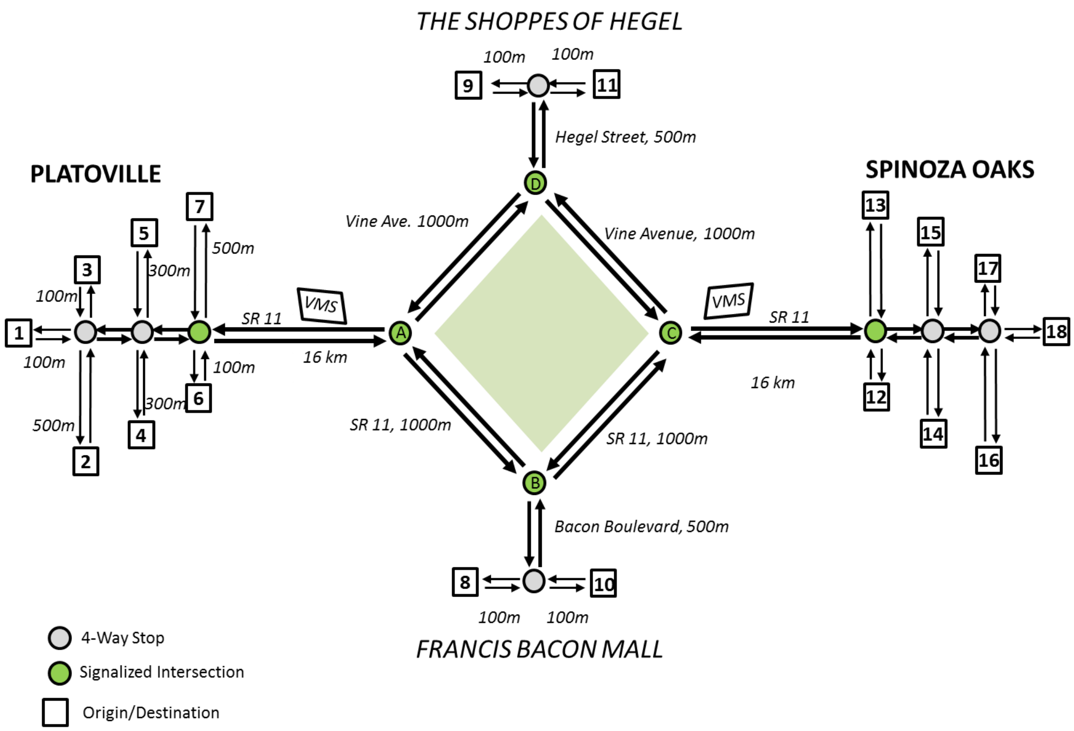

Figure 5-1: Philosopher's Corner VISSIM Test Network

5.2 Variables Examined

The accuracy of the measures estimation was examined by varying the following:

- *Market penetration of connected vehicles*, as our expectation is that market penetration of connected vehicles will increase gradually over time and will vary by vehicle type (e.g., light, transit, freight vehicles) and by region.

- *Traffic conditions* (by varying demand levels, weather impacts, incidents, origin-destinations), to examine accuracy for varying operational conditions.

5.3 Market Penetrations of Connected Vehicles

Measures estimation were performed for three market penetrations of connected vehicles: 2%, 20%, and 95%.

In TCA, the probability that a vehicle is an equipped vehicle capable of transmitting messages, is equal to the specified market penetration. For example, if the market penetration is set as 2%, then in TCA a vehicle has a 2% probability that it is a connected vehicle.

5.4 Traffic Conditions

Measures estimation analysis was performed for four traffic demand levels with varying operational conditions as described in Table 5-1.

Table 5-1: VISSIM and TCA Operational Conditions

Operational Conditions ID	Traffic Demand Multiplier	Number of Incidents	Number of Slippery Conditions
1	Dynamic High	0	1
2	Static Normal	1	1
3	Static High	2	2
4	Dynamic High	2	2

All four operational conditions were tested on the Philosopher's Corner network. The Static High Demand scenario was modeled by increasing the demand by 20% over the entire simulation period. The Dynamic High Demand scenarios was modeled by increasing the demand by 20% over the first hour of simulation and returning to normal demand levels for the last 30 minutes.

For the Philosopher's Corner network, the incidents were modeled as a speed reduction over all lanes for the incident duration and incident area. For the scenarios with a single incident, a reduced speed area affecting all lanes was modeled on the Northeast bound Vine Avenue between points A and D over a 150 foot stretch of roadway. The incident lasted for 30 minutes, starting at 2900 simulation seconds and ending at 4700 simulation seconds. The second incident affected all lanes traveling Eastbound on SR11 between points 1 and A over a 250 foot stretch of roadway. The incident lasted for 20 minutes, starting at 2700 simulation seconds and ending at 3900 simulation seconds. For both incidents, vehicle speeds were reduced to between 2.5 and 3.7 mph.

6 Results

This section discusses results reporting.

6.1 Results Reporting

Results are presented by research hypothesis. Each subsection corresponds to the testing and results of a specific research hypothesis.

For measures estimation exercises, the mean absolute percentage error (MAPE) is reported. The MAPE is a relative measure, which expresses errors as a percentage of the ground truth. MAPE is one of the most often reported performance metrics in transportation since it is intuitive and easy to understand. However, MAPE doesn't reveal the direction of the error (i.e., overestimation or underestimation). Secondly, MAPE tends to favor underestimation rather than overestimation. But due to its simplicity, MAPE is used in this study to report out the performance accuracy for all measures.

6.1.1 Hypothesis 1

Hypothesis: BMM will be able to produce comparable measures estimation results to BSM and CAM.

Results revealed that this hypothesis was supported.

For most market penetrations, the BMM mean absolute percentage errors were within 10% of the results of the best message variant between BSM or CAM, if not better. The graphs below illustrate the MAPE of travel times, queues, turning movements, and slippery conditions estimation for each message type at each market penetration. The MAPE was averaged across the four different operational conditions described in Section 5.4.

For travel time estimation, the BMM results averaged around 12% error across all market penetrations as seen in Figure 6-1 of operational condition ID 2 with static normal vehicle demand, one incident, and one slippery condition region. Travel time estimation errors were within 2% of both BSM and CAM results at each market penetration.

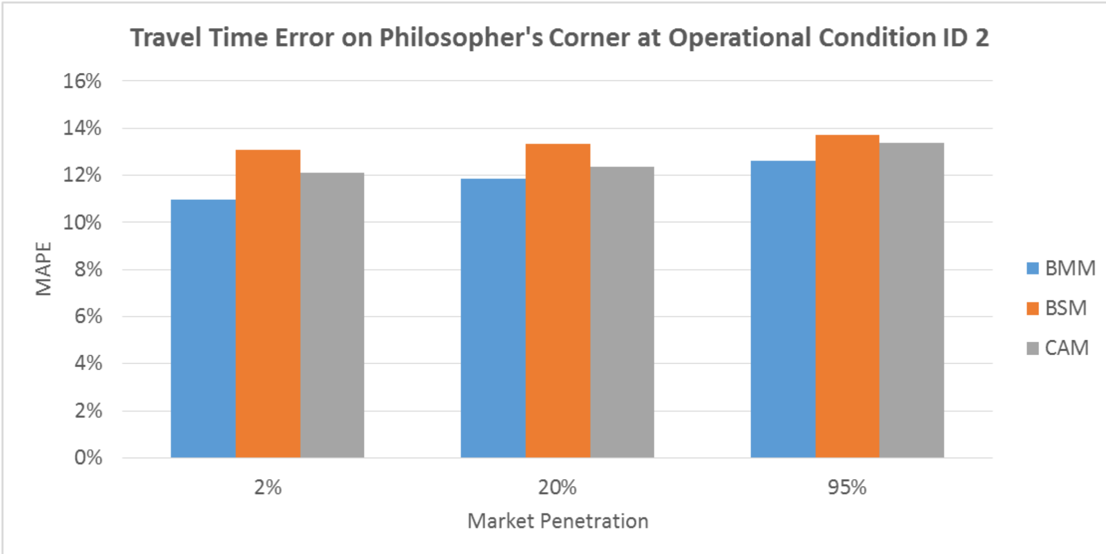

Figure 6-1: Travel Time Estimation Error of BMM, BSM and CAM on Philosopher's Corner at Operational Condition ID 2

Figure 6-2 shows the queue length estimation errors of BMM, BSM, and CAM for Operational Condition ID 2. Errors were nearly identical with an about 90% error rate for all message types at 2% market penetration. Most likely the high errors at low market penetrations were due to the nature of the queue algorithm. The queue algorithm identified a queue only if a vehicle message noted a motionless vehicle within 100 feet of the stop bar. A vehicle is in queue when it is either stopped or is traveling at a speed less than 10 ft/s and is approaching another queued vehicle at headway of less than 100 ft. At low market penetrations, queue lengths were underestimated because there were not enough equipped vehicles for the algorithm to detect a queue, regardless of message type.

At higher market penetrations, the DIDC concept was able to capitalize on the message frequency adjustment capabilities of the DIDC controller. Even though the DIDC concept produced far fewer messages than BSM or CAM as discussed later in Hypothesis 3, the BMM provided enough queue information to produce comparable results to BSM and CAM.

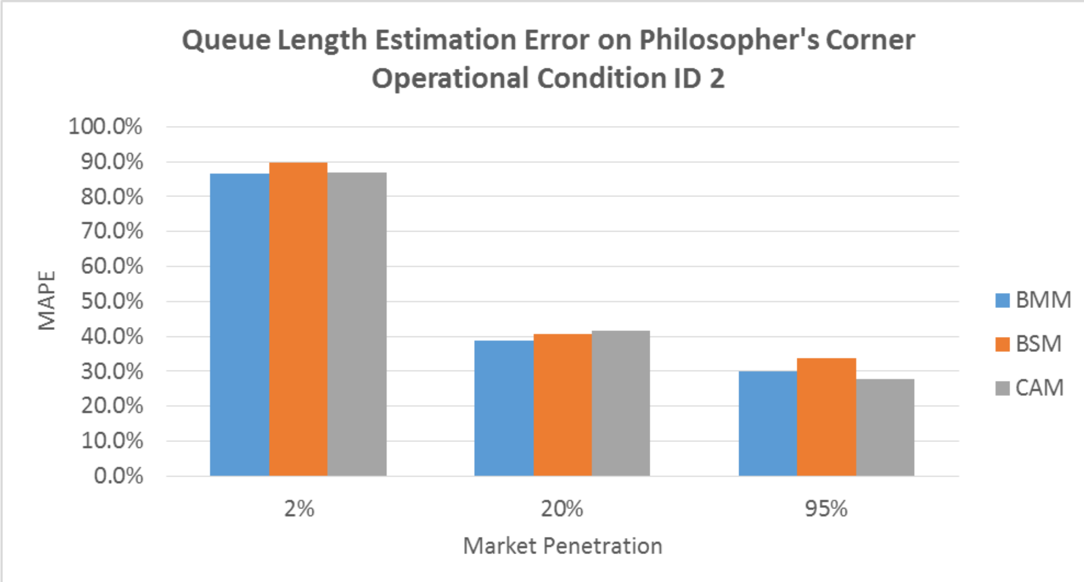

Figure 6-2: Queue Length Estimation Error of BMM, BSM and CAM on Philosopher's Corner Operational Condition ID 2

The turning movement estimation errors for BMMs are within 10% of the BSM and CAM errors as seen in Figure 6-3. Overall these results are comparable. However, the BSM and CAM yielded better turning movements estimations than BMMs at higher market penetrations. This is most likely due to the turning movement algorithm which gives the result as a ratio of left to right turns. Even though the DIDC Controller may have successfully increased or decreased the message yield to match user defined targets for each intersection, more or less data did not improve the ratio result.

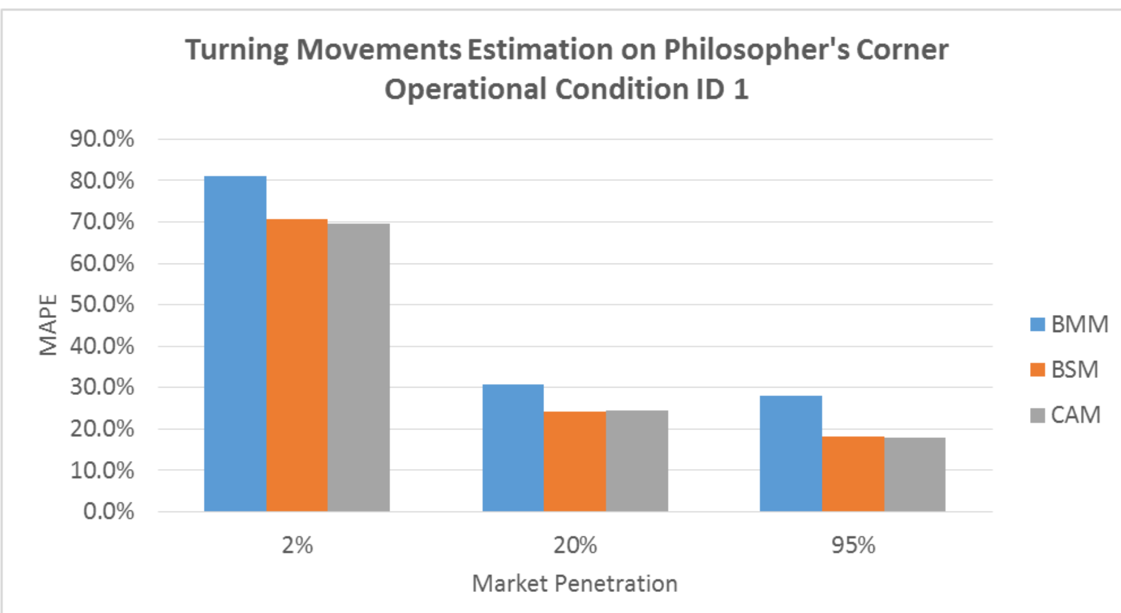

Figure 6-3: Turning Movements Estimation Errors of BMM, BSM and CAM on Philosopher's Corner Operational Condition ID 1

The message elements of the EU CAM do not include vehicle status elements such as traction control so only BSM results could be compared to BMM for slippery conditions. Figure 6-4 shows that the BMM results are comparable to BSM for all market penetrations for Operational Condition 3 which had two slippery condition regions on the Philosopher's Corner network.

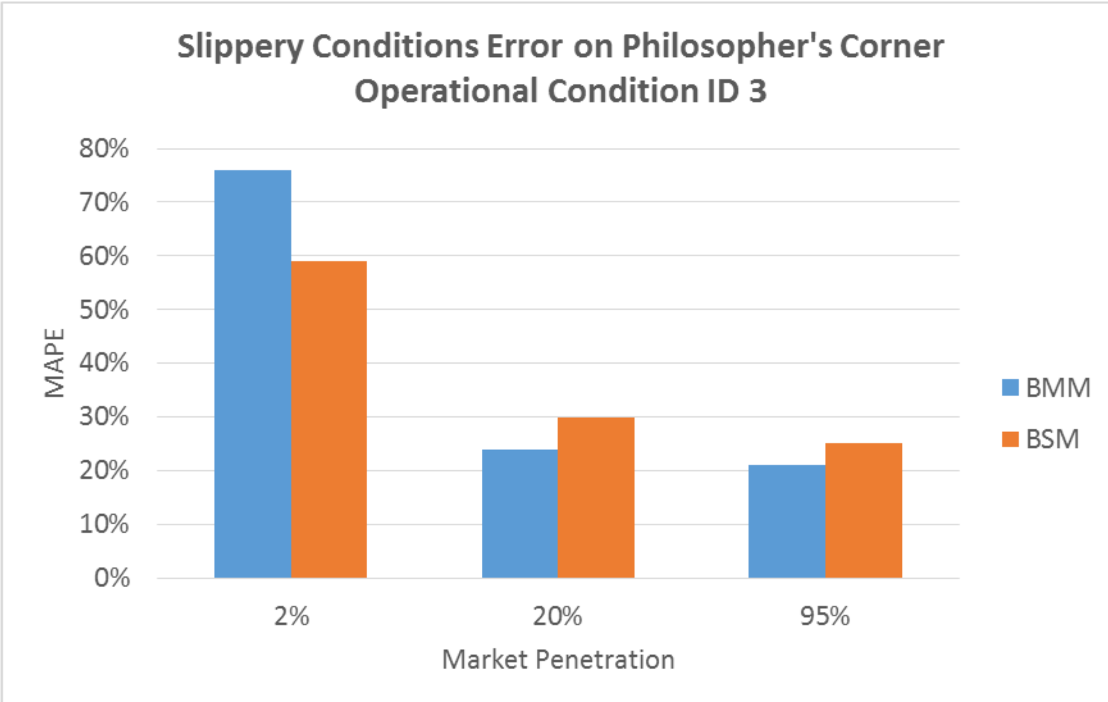

Figure 6-4: Slippery Conditions Estimation Errors of BMM and BSM on Philosopher's Corner Network Operational ID 3

Although BMMs are generated less frequently than BSMs, the DIDC concept was able to produce similar slippery condition accuracy. The DIDC concept utilizes a burst mode concept which allows the DIDC Controller to concentrate vehicle status data collection in areas of interest. In the Philosopher's Corner network, the DIDC Controller enabled burst mode on roadways where a vehicle reported its traction control enabled. When a vehicle reports using traction control, the DIDC Controller notified all vehicle within 100 feet (user-defined parameter) to generate a BMM including traction control information. Figure 6-5 and Figure 6-6 show the data yield of burst mode BMMs compared to the shaded region of how many reported traction control as enabled at both 20% and 95% market penetration for operational condition ID 3 with static high vehicle demand, two incidents, and two slippery conditions. The two slippery condition regions were correctly identified in each simulation.

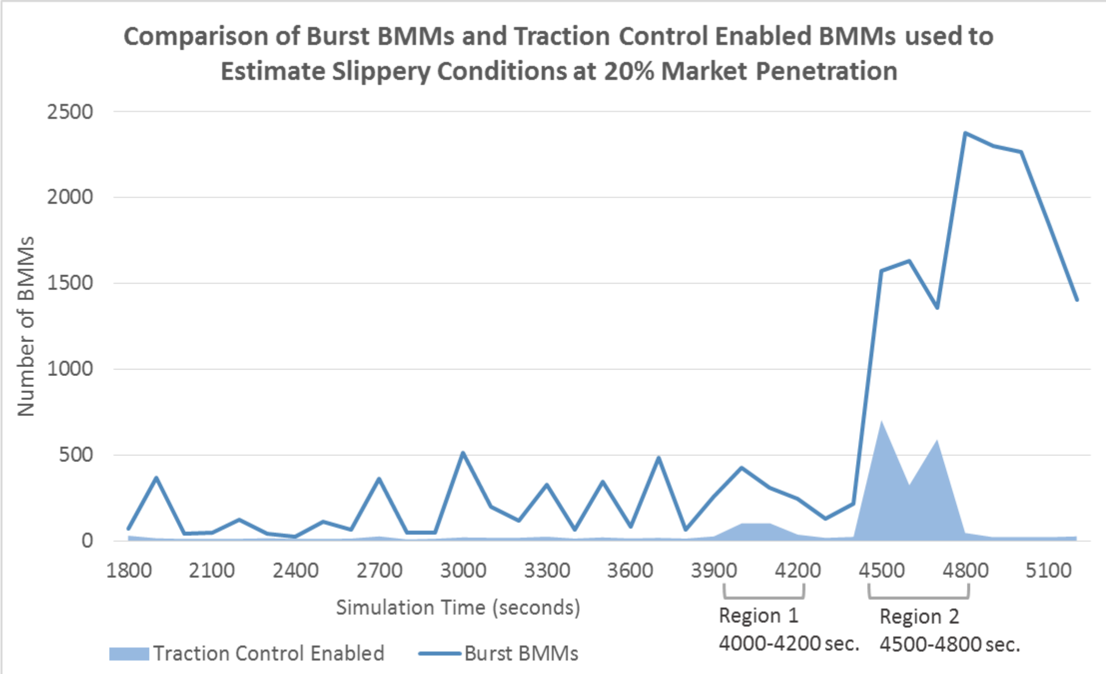

Figure 6-5: Burst BMMs Generated on the Philosopher's Corner Network Operational Condition ID 3 at 20% Market Penetration

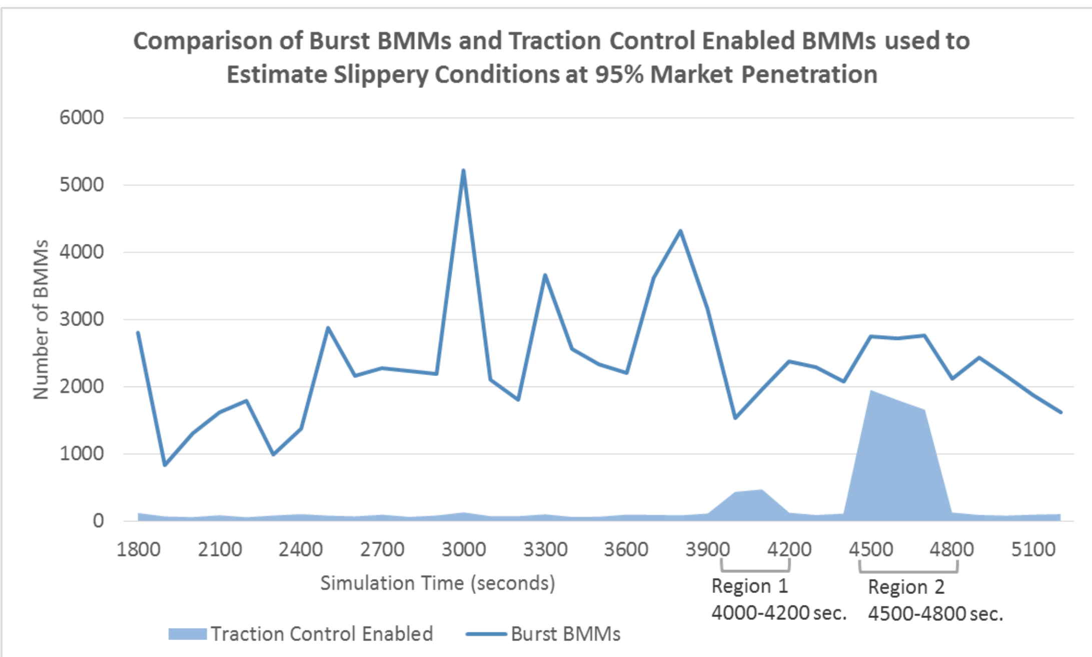

Figure 6-6: Burst BMMs Generated on the Philosopher's Corner Network Operational Condition ID 3 at 95% Market Penetration

6.1.2 Hypothesis 2

Hypothesis: BMM will be able to outperform PDM in measures estimation.

For most market penetrations, the BMM mean absolute percentage error rate was 5-60% better than PDM estimations. Only high market penetrations of turning movements showed the PDM as being more accurate. The graphs below illustrate the MAPE of travel times, queues, turning movements, and slippery conditions estimation for BMM and PDM at each market penetration for operational condition ID 2 with static normal traffic demand, one incident, and one slippery condition.

Travel time estimation results of BMM and PDM are compared in Figure 6-7. While both message types estimated travel times within 30% of ground truth, BMMs did twice as well with error rates under 15% at all market penetration levels.

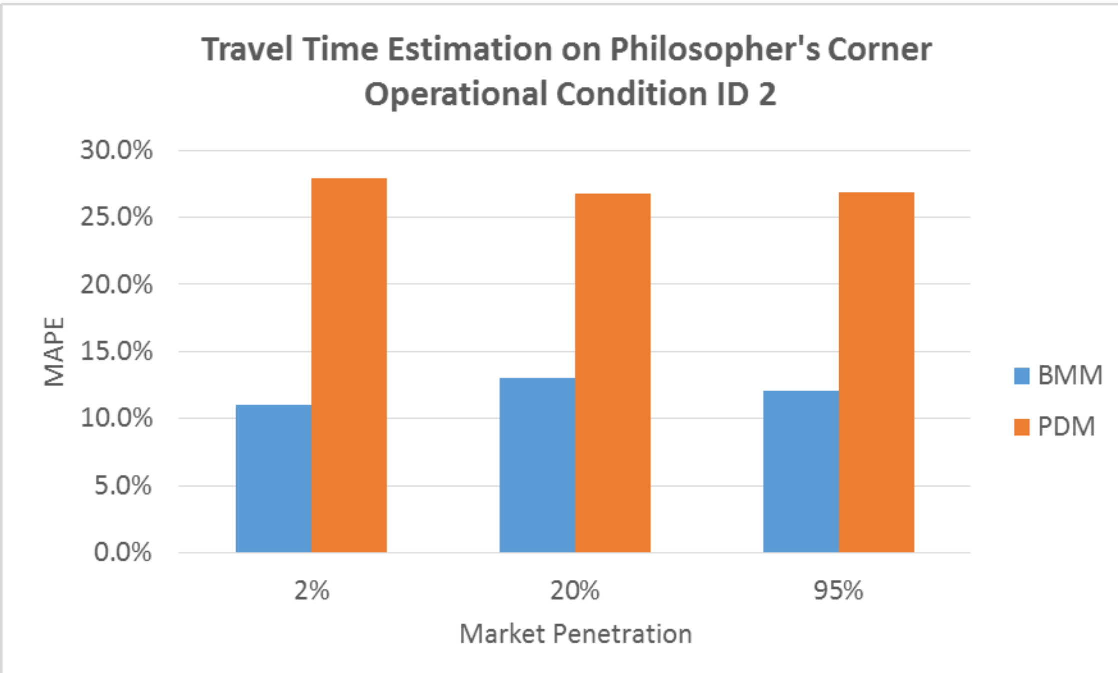

Figure 6-7: Travel Time Estimation MAPE of BMM and PDM on Philosopher's Corner Network Operational ID 2

BMM produced better queue length estimation results as seen in Figure 6-8. A PDM equipped vehicle only generates a vehicle status message when the vehicle stops and starts which may have contributed to the less accurate queue measures results for PDM. In contrast, a queue BMM may be generated more regularly depending on the frequency (lambda) set by the DIDC Controller.

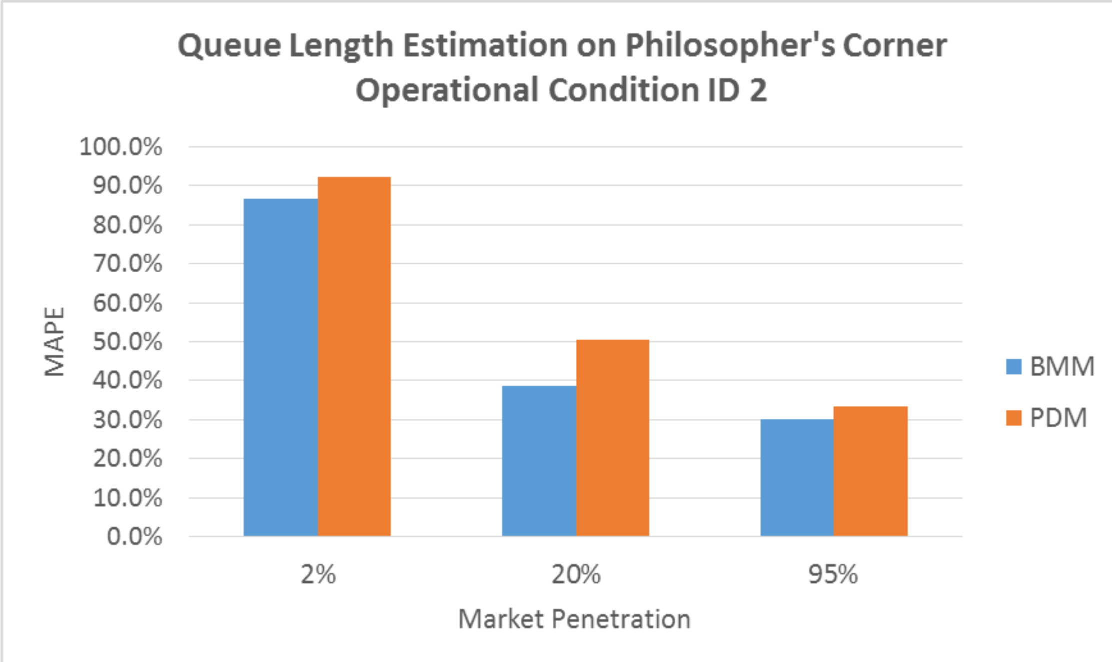

Figure 6-8: Queue Length Estimation MAPE of BMM and PDM on Philosopher's Corner Network Operational ID 2

Turning movement results were more accurate for BMM at 2% and 20% market penetration but PDM results were 6% better than BMM at 95% market penetration. See the explanation for Hypothesis 1 for details on this outcome.

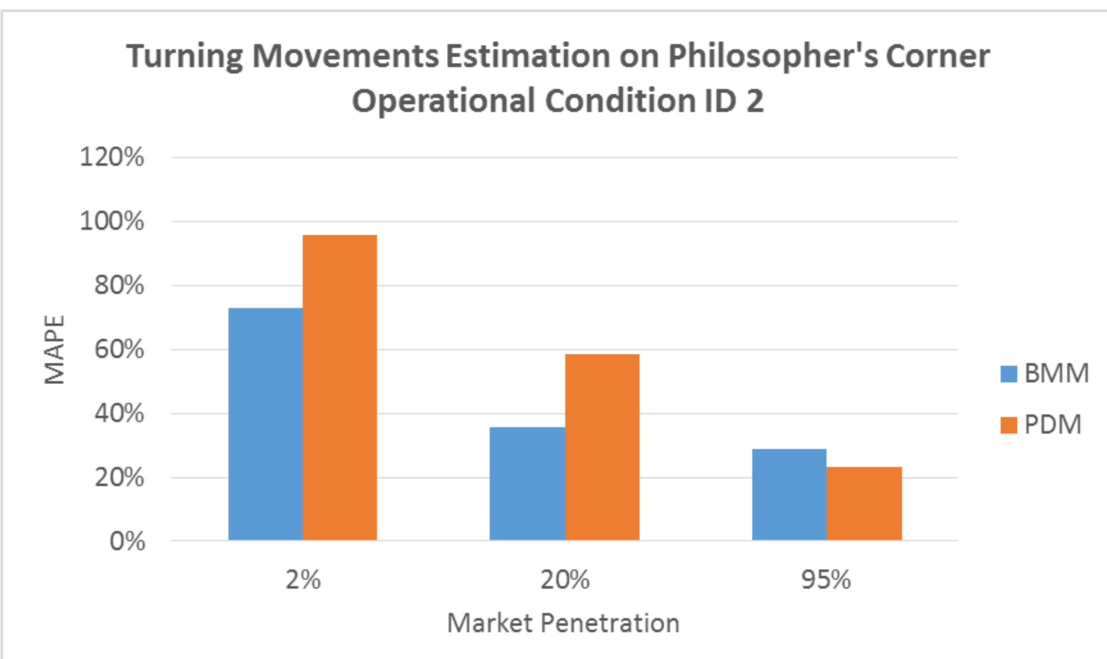

Figure 6-9: Turning Movements Estimation MAPE of BMM and PDM on Philosopher's Corner Network Operational ID 2

The infrequency of PDM generation most likely contributed to their 80% or greater error rates at estimating slippery conditions at 2% and 20% market penetrations. BMMs did not do too much better at 2% market penetration but boast a much improved 20% error at 20% market penetration. The BMM still betters the PDM slippery condition estimation at 95% market penetration but the PDM does drastically improve. This is most likely due to a combination of high market penetration and congestion which provided enough vehicle status information for the slippery condition algorithm to identify the region. A snapshot of the roadway containing the slippery condition is seen in Figure 6-11 to illustrate the congestion.

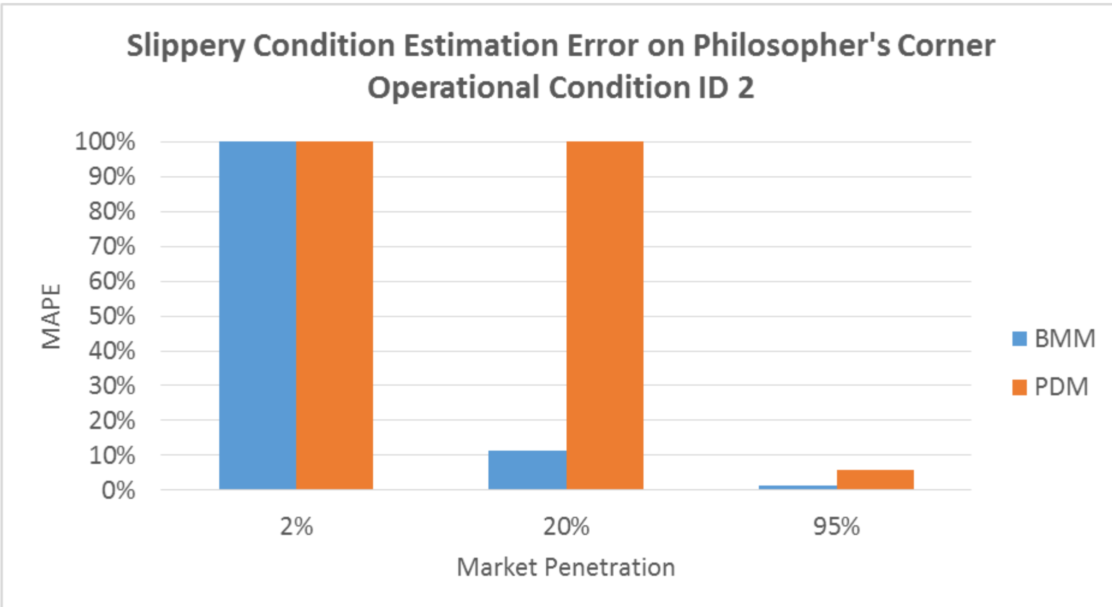

Figure 6-10: Slippery Conditions Estimation MAPE of BMM and PDM on Philosopher's Corner Network Operational ID 2

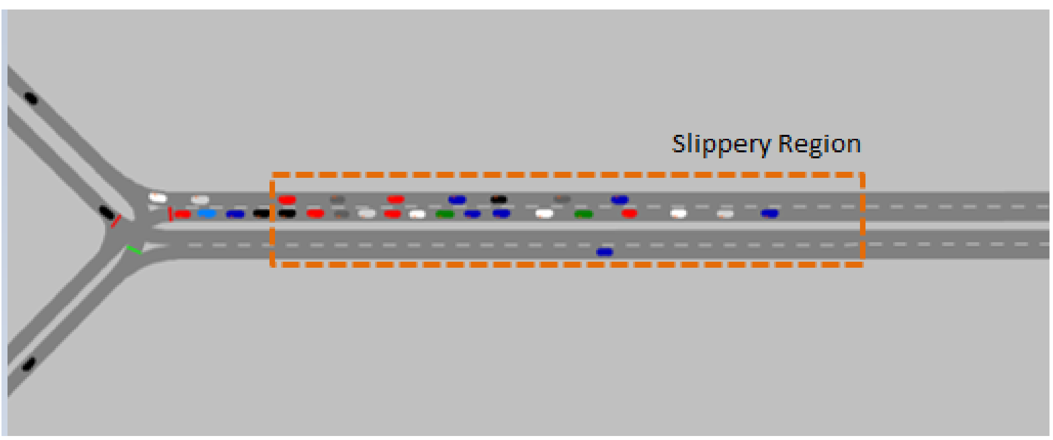

Figure 6-11: Slippery Condition roadway on Philosopher's Corner network

6.1.3 Hypothesis 3

Hypothesis: BMM will be able to produce a lower data load than BSM and CAM.

Results revealed that this hypothesis was supported.

The BSM is a vehicle status message that is generated and transmitted every tenth of a second. The CAM is similar in that it is a vehicle status message that is generated and transmitted no less frequently than every second. As a result, both message types produced a large amount of data as noted in Table 6-1 and illustrated in Figure 6-12. In comparison, a BMM is only generated when the vehicle status or event triggers a series of messages. When a series of BMMs is triggered, the frequency intervals of generation were anywhere from less than a second to over five minutes in the Philosopher's Corner network depending on the trigger type. As a result, the amount of BMMs generated and transmitted were much less than BSM and the EU CAM. Across all market penetrations, BMMs consistently generated 97% less data than BSMs and 88% less data than the EU CAM.

Table 6-1: Average Message Counts for each Market Penetration

	2% Market Penetration	20% Market Penetration	95% Market Penetration
Average Number of BMM	18,419	215,697	867,811
Average Number of BSM	622,877	7,270,729	34,196,782
Average Number of EU CAM	156,856	1,821,096	8,329,690

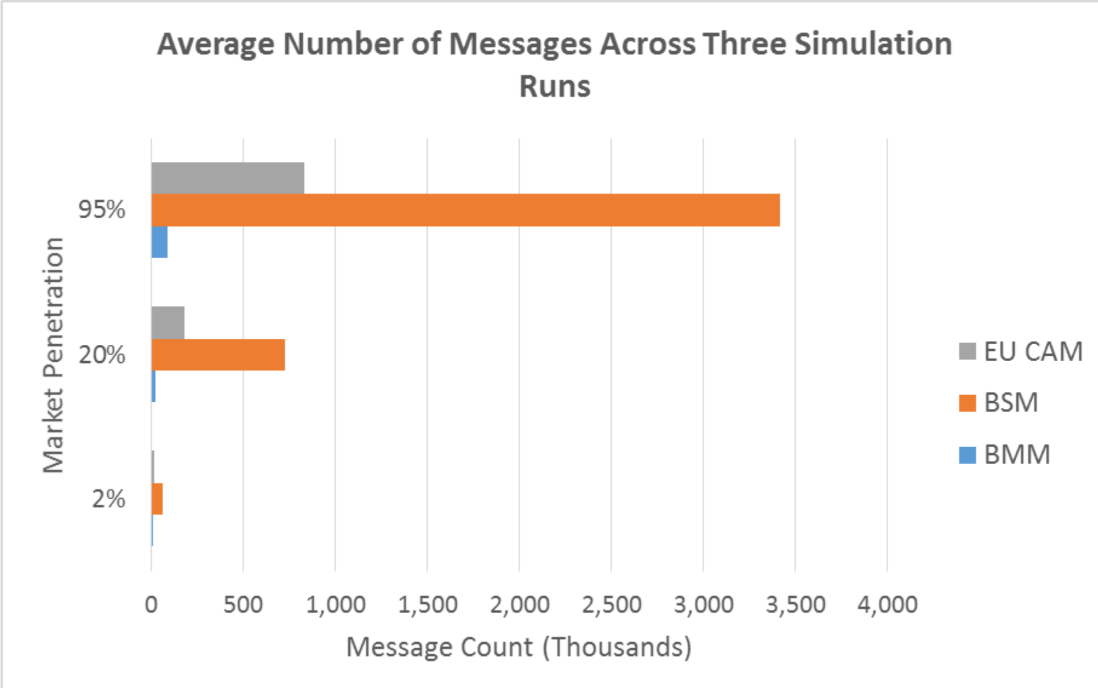

Figure 6-12: Bar chart showing message count differences between BSM, CAM, and BMM

6.1.4 Hypothesis 4

Hypothesis: Vehicle origins and destinations will be easier to re-identify using BSMs than other message variants.

As part of the BSM Data Emulator project, Noblis developed an algorithm that ingests all the available vehicle message data from a simulation run of the Philosopher's Corner network and attempts to identify the origin and destination of five vehicles. Given a single starting message anywhere on the shopping diamond area of the network, the re-identification algorithm works backwards with all the available vehicle data to predict the origin and forwards to predict the destination of the vehicle. Figure 6-13 shows a bar chart illustrating the average percentage of origin-destinations correctly identified across all three rounds of testing. These results include five operational conditions and five different vehicles each round. An origin or destination is correctly identified if the algorithm predicts the correct start or end point on the network **and** the correct vehicle ID. In some cases, the algorithm predicted the correct origin or destination but had the incorrect vehicle ID, these were not counted in the average.

Figure 6-13 confirms the hypothesis that BSMs yielded the most accurate re-identification of vehicle origin and destination. BSM data is every tenth of a second making it easy for the algorithm to track a vehicle along the network, even at 95% market penetration the origin and destination was correctly identified over 95% of the time.

Overall, the re-identification algorithm had relatively good success at identifying the origin and destination across all message types at the 2% market penetration level. Most likely this was due to less vehicles generating and transmitting message in close proximity to each other and therefore making it easier for the algorithm to pick up messages of the correct test vehicle. There is a notable drop in correct identification at the 20% market penetration level where BMM and CAM allow a 40%

success rate which then drops to less than 20% at the 95% market penetration level. The re-identification algorithm had the least success at correctly identifying the origin and destinations using PDM. Most likely this was due to the infrequent message generation rates of PDMs.

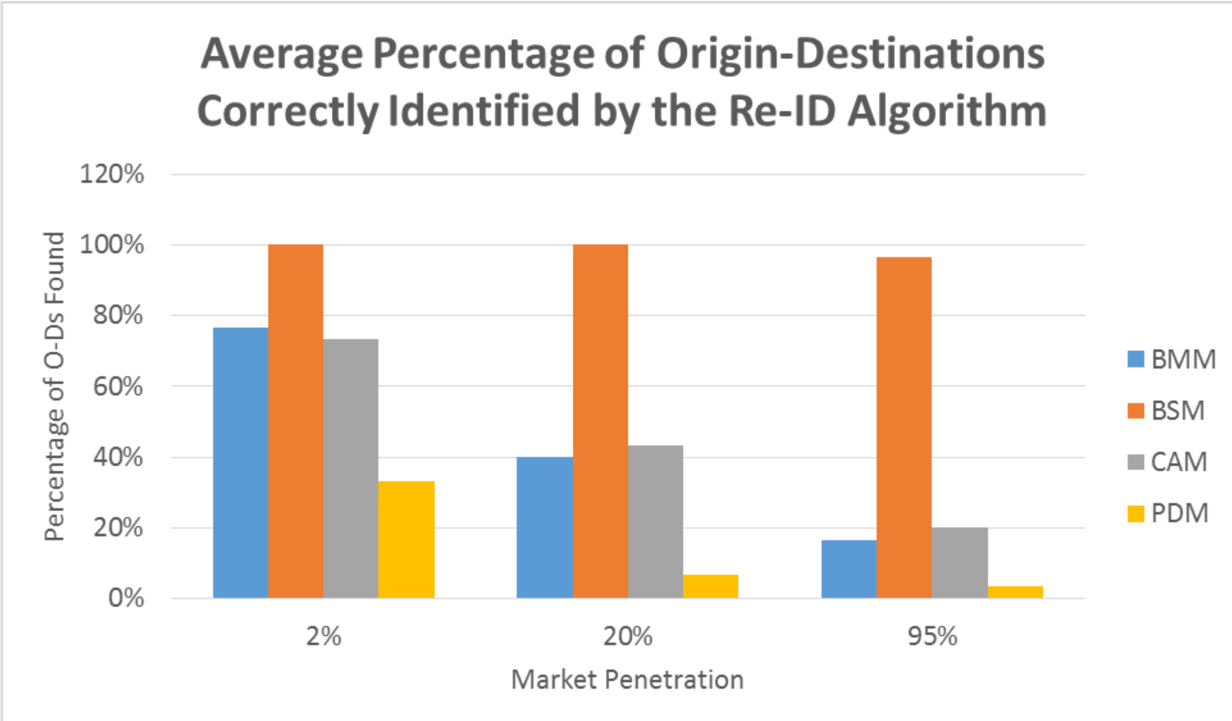

Figure 6-13: Average Percentage of Origin-Destinations Correctly Identified on the Philosopher's Corner network by the Re-ID Algorithm

6.1.5 Hypothesis 5

Hypothesis: The DIDC Controller will be able to adjust BMM generation rates to match the defined targets.

Results reveal that this hypothesis was supported. BMM frequencies at the end of the simulations with 95% market penetration after multiple optimization intervals led to lambda values between 0.2 and 0.6 seconds. The fluctuating targets and generation frequencies (lambda values) of queue BMMs provide an illustration of how well the DIDC Controller was able to adjust to the realized data flow. The target data yield for queue BMMs fluctuates depending on the length of the queue as measured by the DIDC Controller.

Figure 6-14 shows an example of the DIDC Controller's adjustment of queue BMM generation during a model run with 20% market penetration on Philosopher's Corner Operational Condition ID 1 with dynamic high vehicle demand, one incident, and one slippery condition. At the beginning of the simulation, the DIDC Controller consistently reduces the lambda value to increase the frequency of queue BMMs. By the 51 minute mark of the simulation, the realized data yield meets and quickly exceeds the target data yield which causes the lambda values to increase. The corresponding queue length estimation results for this simulation run are seen in Figure 6-15. The linear trend line notes an overall average decrease in MAPE of queue length estimation as the simulation progresses.

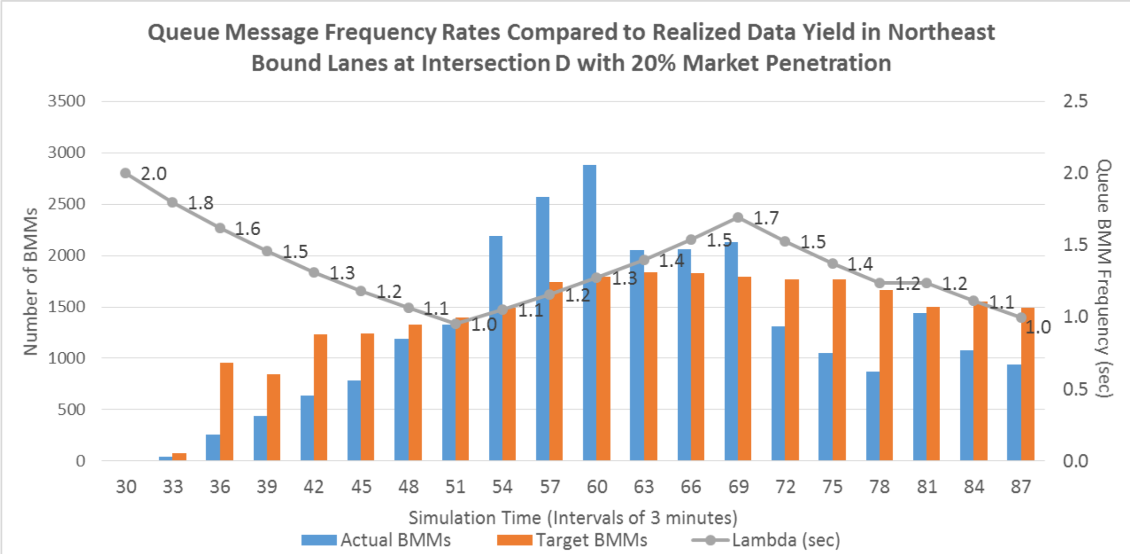

Figure 6-14: DIDC Controller adjustment of queue BMM generation lambda compared to data yield at 20% market penetration

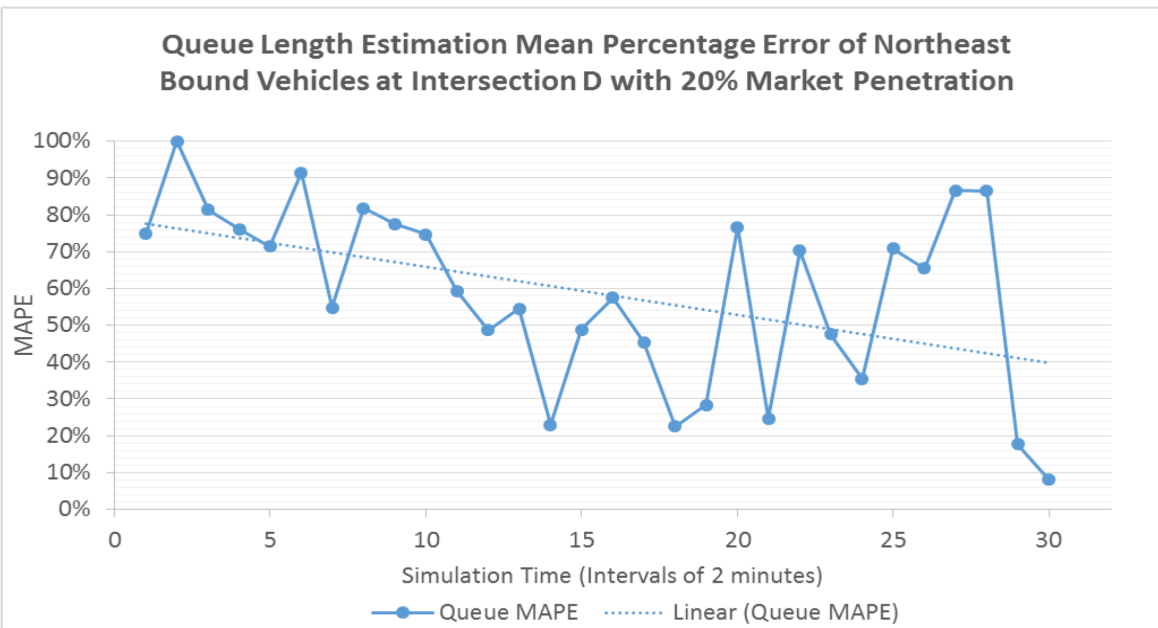

Figure 6-15: BMM queue length estimation error over simulation time at 20% market penetration

Similar graphs in Figure 6-16 and Figure 6-17 show the results for 95% market penetration over the same operational conditions. With a higher market penetration, the DIDC Controller effectively decreases the frequency of queue BMMs by increasing the lambda value between the 30 and 72 simulation minute marks. After 72 minutes, the realized data yield is close enough to the target that the lambda remains constant. A similar linear trend line in Figure 6-17 reveals the same overall improvement to queue length estimation errors as seen in the 20% market penetration example.

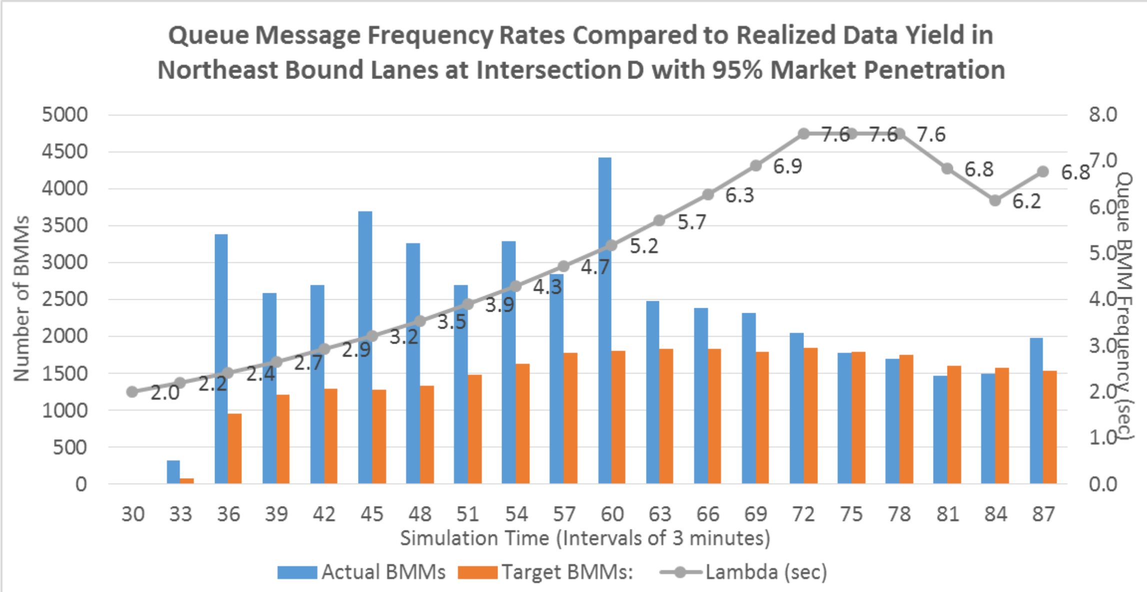

Figure 6-16: DIDC Controller adjustment of queue BMM generation lambda compared to data yield at 95% market penetration

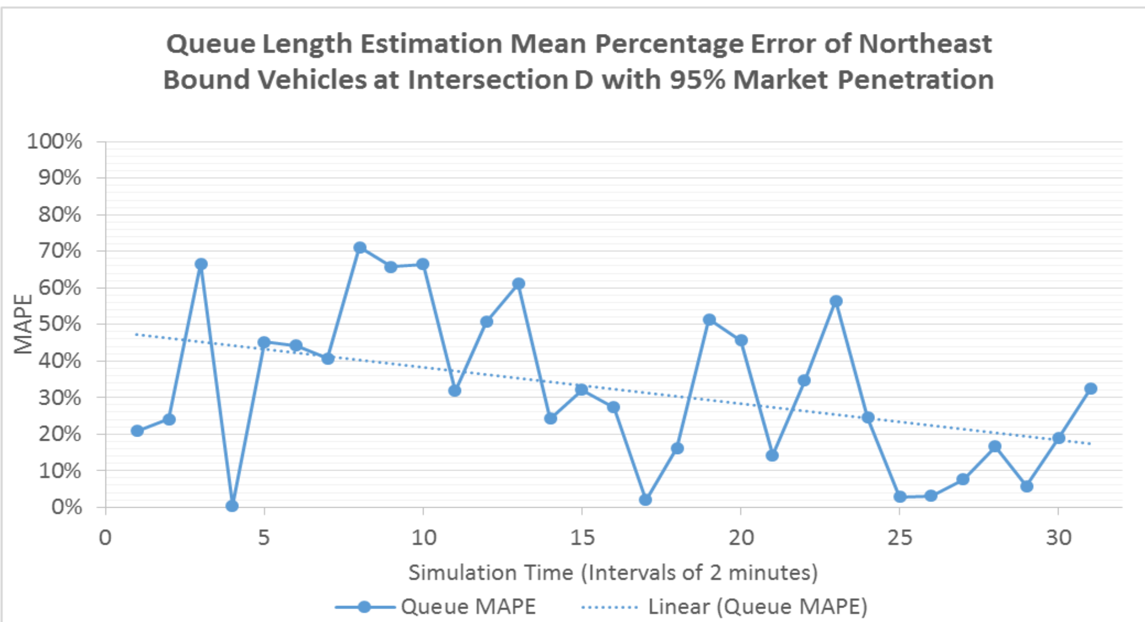

Figure 6-17: BMM queue length estimation error over simulation time at 95% market penetration

7 Conclusions

One of the overarching purposes of DIDC was to determine if there was a logical way to optimize the torrent of connected vehicle data in a way that was feasible and useful. Two ways connected vehicle data could be optimized include: alter the vehicle data generation rates from a probe data perspective or filter unneeded or redundant data from a roadside perspective. The perspective of DIDC is to investigate both ways to make connected vehicle data as efficient and effective as possible by dynamically modifying vehicle data generation rates in response to current data needs and actual data yield from the transportation system. Key findings from the testing that was conducted to study and compare the DIDC concept with existing vehicle message processes is included in this section.

7.1 Key Findings

- **The DIDC concept generated data more efficiently than other message types**: The DIDC concept effectively throttled up data in low market penetration simulations and reduced redundant data in high market penetrations without sacrificing measures estimation capabilities for travel times, queues, turning movements, and slippery conditions.

- **The DIDC concept produced BMMs that were harder to re-identify the vehicle origin and destination compared to BSM and CAM, but easier than PDM:** A vehicle's origin and destination could be re-identified less than 35% of the time using BMMs at market penetrations greater than 20% on the Philosopher's Corner network. By comparison, the origin and destination could be re-identified more than 90% of the time using BSMs but less than 20% of the time using PDMs.

- **The TCA Version 2.4 software tool provides capability for others to build upon and find better ways of optimization:** The testing documented and completed in the BSM Emulator project looked at simple adaptive processes primarily to see if the measures estimation results were preserved or improved while reducing data flow. There is apparent potential value in these types of adaptive processes based on the research outlined in this report. Much more research could be explored in this area and the TCA Version 2.4 tool provides a good starting point for continuing investigation.

References

1. Society of Automotive Engineers (SAE) standard J2735, Dedicated Short Range Communications (DSRC) Message Set Dictionary, November 2009.
2. Deurbrouck, T., Larkin, J., and K. Wunderlich. Trajectory Conversion Algorithm (TCA) Software, Version 2: Concept of Operations, Prepared by Noblis for U.S.DOT, February 2016.
3. Dowling, R. Volume VI: Definition, Interpretation, And Calculation Of Traffic Analysis Tools Measures of Effectiveness, FHWA-HOP-08-054, Prepared by Cambridge Systematics for U.S.DOT, January 2007.
4. Vasudevan, M., O'Hara, J., Ozbay, K., Mudigonda, S., and A. Kurkcu. Estimating Transportation Measures Using Connected Vehicle Messages, Prepared by Noblis for U.S. DOT, February 2016.

APPENDIX A. List of Acronyms

Acronym	Meaning
AERIS	Applications for the Environment: Real-Time Information Synthesis
BMM	Basic Mobility Message
BSM	Basic Safety Message
CAM	Cooperative Awareness Message
DCM	Data Capture and Management
DIDC	Dynamic Interrogative Data Capture
DMA	Dynamic Mobility Applications
DOT	Department of Transportation
DSRC	Dedicated Short Range Communications
DSS	Decision Support System
FHWA	Federal Highway Administration
ITS	Intelligent Transportation Systems
JPO	Joint Program Office
PDM	Probe Data Message
RITA	Research and Innovative Technology Administration
TCA	Trajectory Conversion Algorithm
TRB	Transportation Research Board
USDOT	U.S. Department of Transportation
VII	Vehicle Infrastructure Integration

APPENDIX B. Re-Identification Algorithm

BSM Re-Identification Algorithm

Notations

T	=	Simulation duration
t	=	Instantaneous time, where $t \in T$
B	=	Array of known starting BSMs
b	=	ID for known starting BSM, where $b \in B$
d	=	ID for temporary current BSM moving toward destination
o	=	ID for temporary current BSM moving toward origin
p_b	=	Initial position of starting BSM, b
L_b	=	Initial link of starting BSM, b
t_b	=	Time of starting BSM, b
l_b	=	Length of starting BSM, b
w_b	=	Width of starting BSM, b
L	=	ID for roadway segment (i.e., link)
$\overline{L}$	=	Set of all roadway segments (links) L in the network (i.e., $L \in \overline{L}$)
$A(L)$	=	List of all links that are adjacent to link L
$I(L,t)$	=	List of all Basic Safety Messages (BSM) generated at time t, on link L
i	=	BSM, where $i \in I(L,t)$
t_i	=	Time stamp of BSM i
x_i	=	Position associated with BSM i
l_i	=	Length associated with BSM i
w_i	=	Width associated with BSM i
v_i	=	Speed associated with BSM i
a_i	=	Acceleration associated with BSM i
$DIS(x_i, x_j)$	=	Function that measures in feet the distance between message i and message j (Initialize to 0)
R	=	Set of all routes in the network (Each route comprises a route origin, r_O and destination, r_D)
$O(b)$	=	List of BSMs that go from the starting BSM, b to the origin of the route, $r_O \in R$
$D(b)$	=	List of BSMs that go from the starting BSM, b to the destination of the route, $r_D \in R$
$E(b,i)$	=	Function that measures the error between message i and message b
$ID(b)$	=	List of BSMs that go from origin to destination of route for initial starting BSM, b
$ABS()$	=	Function that returns the absolute value

Process

Step 1: For $b \in B$ do steps 2-20.

Step 2: Select starting BSM, b.

Step 3: For all BSMs generated on the network for all times, $I(\overline{L}, T)$, select all $i \in I(\overline{L}, T)$, where $l_i = l_b$ and $w_i = w_b$.

Step 4: Set $d = b$. Add b to the destination list $D(b)$. Repeat steps 5-11 until r_D has been found.

Step 5: For each link that could be travelled on from the initial BSM link, $L \in A(L_d)$, find all BSMs on those links at time $(t_d + 0.1\ seconds)$, $I(L, t_d + 0.1)$.

Step 6: For each BSM, $i \in I(L, t_d + 0.1)$, do steps 7-10.

Step 7: If x_i is upstream of p_d return to Step 6.

Step 8: Calculate the distance between i and d, $DIS(p_d, x_i)$.

Step 9: Calculate the estimated distance the vehicle traveled towards the destination from initial position, $DIS(p_d, x_D)$ using the following:

$$v_d * 0.1 + 0.5 * a_d * 0.1^2$$

Step 10: Calculate the error, $E(d, i)$ as follows:

$$ABS(DIS(p_d, x_i) - DIS(p_d, x_D))$$

Step 11: Select the BSM, i, with the lowest error, $E(d, i)$, add it to the destination list $D(b)$, set $d = i$ and return to Step 5. If $I(L, t_d + 0.1)$ is empty, r_D has been found, proceed to Step 12.

Step 12: Set $o = b$. Repeat steps 13-19 until r_O has been found.

Step 13: For each link that could be travelled on from the initial BSM link, $L \in A(L_o)$, find all BSMs on those links at time $(t_o - 0.1\ seconds)$, $I(L, t_o - 0.1)$.

Step 14: For each BSM, $i \in I(L, t_o - 0.1)$, do steps 15-18.

Step 15: If x_i is downstream of p_o return to Step 14.

Step 16: Calculate the distance between i and , $DIS(p_o, x_i)$.

Step 17: Calculate the estimated distance the vehicle traveled towards the initial position from the new position, $DIS(x_i, x_D)$ using the following:

$$v_i * 0.1 + 0.5 * a_i * 0.1^2$$

Step 18: Calculate the error, $E(o, i)$ as follows:

$$ABS(DIS(p_o, x_i) - DIS(x_i, x_D))$$

Step 19: Select the BSM, i, with the lowest error, $E(o, i)$, add it to the origin list $O(b)$, set $o = i$ and return to Step 13. If $I(L, t_o - 0.1)$ is empty, r_O has been found, proceed to Step 20.

Step 20: Merge the origin list, $O(b)$, and the destination list $D(b)$, to form the full re-identified route, $ID(b)$. Print $ID(b)$. Return to Step 1. If all initial BSMs have been re-identified, terminate.

CAM Re-Identification Algorithm

Notations

T	=	Simulation duration
t	=	Instantaneous time, where $t \in T$
B	=	Array of known starting CAMs
b	=	ID for known starting CAM, where $b \in B$
d	=	ID for temporary current CAM moving toward destination
o	=	ID for temporary current CAM moving toward origin
p_b	=	Initial position of starting CAM, b
L_b	=	Initial link of starting CAM, b
t_b	=	Time of starting CAM, b
l_b	=	Length of starting CAM, b
w_b	=	Width of starting CAM, b
L	=	ID for roadway segment (i.e., link)
$\overline{L}$	=	Set of all roadway segments (links) L in the network (i.e., $L \in \overline{L}$)
$A(L)$	=	List of all links that are adjacent to link L
$I(L,t)$	=	List of all Cooperative Awareness Messages (CAM) generated at time t, on link L
i	=	CAM, where $i \in I(L,t)$
t_i	=	Time stamp of CAM i
x_i	=	Position associated with CAM i
l_i	=	Length associated with CAM i
w_i	=	Width associated with CAM i
v_i	=	Speed associated with CAM i
a_i	=	Acceleration associated with CAM i
h_i	=	Heading associated with CAM i
$DIS(x_i, x_j)$	=	Function that measures in feet the distance between message i and message j (Initialize to 0)
R	=	Set of all routes in the network (Each route comprises a route origin, r_O and destination, r_D)
$O(b)$	=	List of CAMs that go from the starting CAM, b to the origin of the route, $r_O \in R$
$D(b)$	=	List of CAMs that go from the starting CAM, b to the destination of the route, $r_D \in R$
$E(b,i)$	=	Function that measures the error between message i and message b
$ID(b)$	=	List of CAMs that go from origin to destination of route for initial starting CAM, b
$HEAD(h_b, h_i)$	=	Function that measures the change in heading between message i and message b
$SPD(v_b, v_i)$	=	Function that measures the change in speed between message i and message b
$ABS()$	=	Function that returns the absolute value

Process

Step 1: For $b \in B$ do steps 2-22.

Step 2: Select starting CAM, b.

Step 3: For all CAMs generated on the network for all times, $I(\bar{L},T)$, select all $i \in I(\bar{L},T)$, where $l_i = l_b$ and $w_i = w_b$.

Step 4: Set $d = b$. Add b to the destination list $D(b)$. Repeat steps 5-12 until r_D has been found.

Step 5: For each link that could be travelled on from the initial CAM link, $L \in A(L_d)$, find all CAMs on those links between times t_d and $(t_d + 1.0\ seconds)$, $I(L,([t_d,t_d + 1.0])$.

Step 6: For each CAM, $i \in I(L,([t_d,t_d + 1.0])$, do steps 7-11.

Step 7: If x_i is upstream of p_d return to Step 6.

Step 8: If $t_i < t_d + 1.0$ check if any CAM triggers have been met using the following conditions:

If $DIS(p_d,x_i) > 13.124$ ft or $SPD(v_d,v_i) > 1.11847$ ft/sec or $HEAD(h_d,h_i) > 4$ degrees continue to step 29. Otherwise, return to Step 6.

Step 9: Calculate the distance between i and d, $DIS(p_d,x_i)$.

Step 10: Calculate the estimated distance the vehicle traveled towards the destination from initial position, $DIS(p_d,x_D)$ using the following:

$$v_d * (t_i - t_d) + 0.5 * a_d * (t_i - t_d)^2$$

Step 11: Calculate the error, $E(d,i)$ as follows:

$$ABS(DIS(p_d,x_i) - DIS(p_d,x_D))$$

Step 12: Select the CAM, i, with the lowest error, $E(d,i)$, add it to the destination list $D(b)$, set $d = i$ and return to Step 5. If $I(L,([t_d,t_d + 1.0])$ is empty, r_D has been found, proceed to Step 13.

Step 13: Set $o = b$. Repeat steps 13-21 until r_O has been found.

Step 14: For each link that could be travelled on from the initial CAM link, $L \in A(L_o)$, find all CAMs on those links between times t_d and $(t_d - 1.0\ seconds)$, $I(L,([t_d,t_d - 1.0])$.

Step 15: For each CAM, $i \in I(L,([t_d,t_d - 1.0])$, do steps 16-20.

Step 16: If x_i is downstream of p_o return to Step 14.

Step 17: If $t_i < t_d + 1.0$ check if any CAM triggers have been met using the following conditions:

If $DIS(p_d,x_i) > 13.124$ ft or $SPD(v_d,v_i) > 1.11847$ ft/sec or $HEAD(h_d,h_i) > 4$ degrees continue to step 29. Otherwise, return to Step 14.

Step 18: Calculate the distance between i and , $DIS(p_o,x_i)$.

Step 19: Calculate the estimated distance the vehicle traveled towards the initial position from the new position, $DIS(x_i, x_D)$ using the following:

$$v_i * (t_d - t_i) + 0.5 * \ a_i * \ (t_d - t_i)^2$$

Step 20: Calculate the error, $E(o, i)$ as follows:

$$ABS(DIS(p_o, x_i) - DIS(x_i, x_D))$$

Step 21: Select the CAM, i, with the lowest error, $E(o, i)$, add it to the origin list $O(b)$, set $o = i$ and return to Step 14. If $I(L, ([t_d, t_d - 1.0])$ is empty, r_O has been found, proceed to Step 22.

Step 22: Merge the origin list, $O(b)$, and the destination list $D(b)$, to form the full re-identified route, $ID(b)$. Print $ID(b)$. Return to Step 1. If all initial CAMs have been re-identified, terminate.

BMM Re-Identification Algorithm

Notations

T	=	Simulation duration
t	=	Instantaneous time, where $t \in T$
B	=	Array of known starting BMMs
b	=	ID for known starting BMM, where $b \in B$
d	=	ID for temporary current BMM moving toward destination
o	=	ID for temporary current BMM moving toward origin
p_b	=	Initial position of starting BMM, b
L_b	=	Initial link of starting BMM, b
t_b	=	Time of starting BMM, b
l_b	=	Length of starting BMM, b
w_b	=	Width of starting BMM, b
L	=	ID for roadway segment (i.e., link)
$\overline{L}$	=	Set of all roadway segments (links) L in the network (i.e., $L \in \overline{L}$)
$A(L)$	=	List of all links that are adjacent to link L
$I(L,t)$	=	List of all Basic Mobility Messages (BMM) generated at time t, on link L
i	=	BMM, where $i \in I(L,t)$
t_i	=	Time stamp of BMM i
x_i	=	Position associated with BMM i
l_i	=	Length associated with BMM i
w_i	=	Width associated with BMM i
v_i	=	Speed associated with BMM i
a_i	=	Acceleration associated with BMM i
mt_i	=	The message type of BMM i
$DIS(x_i, x_j)$	=	Function that measures in feet the distance between message i and message j (Initialize to 0)
R	=	Set of all routes in the network (Each route comprises a route origin, r_O and destination, r_D)
$O(b)$	=	List of BMMs that go from the starting BMM, b to the origin of the route, $r_O \in R$
$D(b)$	=	List of BMMs that go from the starting BMM, b to the destination of the route, $r_D \in R$
$E(b,i)$	=	Function that measures the error between message i and message b
$ID(b)$	=	List of BMMs that go from origin to destination of route for initial starting BMM, b
$MGTS(\overline{L},t)$	=	List of times, t, when a new Mean Generation Time for BMMs was sent out to the network, $\overline{L}$.
m	=	Current Mean Generation Time for BMMs, where $m \in MGTS(\overline{L})$
n	=	Estimated next periodic time
$ABS()$	=	Function that returns the absolute value

Process

Step 1: For $b \in B$ do steps 2-24.

Step 2: Select starting BMM, b.

Step 3: For all BMMs generated on the network for all times, $I(\overline{L}, T)$, select all $i \in I(\overline{L}, T)$, where $l_i = l_b$ and $w_i = w_b$.

Step 4: Set $d = b$. Add b to the destination list $D(b)$. Set m equal to initial value in $MGTS(\overline{L}, t)$ where $t > t_d$. Set n equal to $t_d + m$. Repeat steps 5-13 until r_D has been found.

Step 5: For each link that could be travelled on from the initial BMM link, $L \in A(L_d)$, find all BMMs on those links between times t_d and $(n + 2 * m)$, $I(L, [t_d, (n + 2 * m)])$.

Step 6: For each BMM, $i \in I(L, [t_d, (n + 2 * m)])$, do steps 7-10.

Step 7: If x_i is upstream of p_d return to Step 6.

Step 8: Calculate the distance between i and d, $DIS(p_d, x_i)$.

Step 9: Calculate the estimated distance the vehicle traveled towards the destination from initial position, $DIS(p_d, x_D)$ using the following:

$$.5 * ((v_d * (t_i - t_d) + 0.5 * a_d * (t_i - t_d)^2) + (v_d * (t_i - t_d)))$$

Step 10: Calculate the error, $E(d, i)$ as follows:

$$ABS(DIS(p_d, x_i) - DIS(p_d, x_D))$$

If message i is a periodic, $mt_i = 1$, then add the following to the error, $E(d, i)$:

$$ABS(n - t_i)$$

Step 11: Select the BMM, i, with the lowest error, $E(d, i)$, add it to the destination list $D(b)$, set $d = i$. If $I(L, [t_d, (n + 2 * m)])$ is empty, r_D has been found, proceed to Step 14.

Step 12: Check if t_d is greater than the next time in $MGTS(\overline{L})$. If so set m equal to new Mean Generation Time.

Step 13: If new message, d, is a periodic, $mt_d = 1$, or $t_d > n$ set n equal to $t_d + m$. Return to Step 5.

Step 14: Set $o = b$. Set m equal to initial value in $MGTS(\overline{L}, t)$ where $t \leq t_o$. Set n equal to $t_o - m$. Repeat steps 15-23 until r_O has been found.

Step 15: For each link that could be travelled on from the initial BMM link, $L \in A(L_o)$, find all BMMs on those links between times t_o and $(n - 2 * m)$, $I(L, [t_o, (n - 2 * m)])$.

Step 16: For each BMM, $i \in I(L, [t_o, (n - 2 * m)])$, do steps 17-20.

Step 17: If x_i is downstream of p_o return to Step 16.

Step 18: Calculate the distance between i and , $DIS(p_o, x_i)$.

Step 19: Calculate the estimated distance the vehicle traveled towards the initial position from the new position, $DIS(x_i, x_D)$ using the following:

$$.5 * ((v_i * (t_i - t_o) + 0.5 * a_i * (t_i - t_o)^2) + (v_i * (t_i - t_o)))$$

Step 20: Calculate the error, $E(o, i)$ as follows:

$$ABS(DIS(p_o, x_i) - DIS(x_i, x_D))$$

If message i is a periodic, $mt_i = 1$, then add the following to the error, $E(d, i)$:

$$ABS(n - t_i)$$

Step 21: Select the BMM, i, with the lowest error, $E(o, i)$, add it to the origin list $O(b)$, set $o = i$ and return to Step 14. If $I(L, [t_o, (n - 2 * m)])$ is empty, r_O has been found, proceed to Step 24.

Step 22: Check if t_d is less than the previous time in $MGTS(\bar{L})$. If so set m equal to new Mean Generation Time.

Step 23: If new message, d, is a periodic, $mt_d = 1$, or $t_d < n$ set n equal to $t_d - m$. Return to step 15.

Step 24: Merge the origin list, $O(b)$, and the destination list $D(b)$, to form the full re-identified route, $ID(b)$. Print $ID(b)$. Return to Step 1. If all initial BMMs have been re-identified, terminate.

PDM Re-Identification Algorithm

Notations

T	=	Simulation duration
t	=	Instantaneous time, where $t \in T$
B	=	Array of known starting PDMs
b	=	ID for known starting PDM, where $b \in B$
d	=	ID for temporary current PDM moving toward destination
o	=	ID for temporary current PDM moving toward origin
p_b	=	Initial position of starting PDM, b
L_b	=	Initial link of starting PDM, b
t_b	=	Time of starting PDM, b
l_b	=	Length of starting PDM, b
w_b	=	Width of starting PDM, b
L	=	ID for roadway segment (i.e., link)
$\overline{L}$	=	Set of all roadway segments (links) L in the network (i.e., $L \in \overline{L}$)
$A(L)$	=	List of all links that are adjacent to link L
$I(L,t)$	=	List of all Probe Data Messages (PDM) generated at time t, on link L
i	=	PDM, where $i \in I(L,t)$
t_i	=	Time stamp of PDM i
x_i	=	Position associated with PDM i
l_i	=	Length associated with PDM i
w_i	=	Width associated with PDM i
v_i	=	Speed associated with PDM i
a_i	=	Acceleration associated with PDM i
$DIS(x_i, x_j)$	=	Function that measures in feet the distance between message i and message j (Initialize to 0)
R	=	Set of all routes in the network (Each route comprises a route origin, r_O and destination, r_D)
$O(b)$	=	List of PDMs that go from the starting PDM, b to the origin of the route, $r_O \in R$
$D(b)$	=	List of PDMs that go from the starting PDM, b to the destination of the route, $r_D \in R$
$E(b,i)$	=	Function that measures the error between message i and message b
$ID(b)$	=	List of PDMs that go from origin to destination of route for initial starting PDM, b
$ABS()$	=	Function that returns the absolute value
n	=	Estimated next periodic time
mt_i	=	Message type of message i

Process

Step 1: For $b \in B$ do steps 2-20.

Step 2: Select starting PDM, b.

Step 3: For all PDMs generated on the network for all times, $I(\bar{L}, T)$, select all $i \in I(\bar{L}, T)$, where $l_i = l_b$ and $w_i = w_b$.

Step 4: Set $d = b$. Add b to the destination list $D(b)$. Repeat steps 5-11 until r_D has been found.

Step 5: For each link that could be travelled on from the initial PDM link, $L \in A(L_d)$, find all PDMs on those links between time t_d and , $I(L, [t_d, n])$.

Step 6: For each PDM, $i \in I(L, [t_d, n])$do steps 7-10.

Step 7: If x_i is upstream of p_d return to Step 6.

Step 8: Find all messages, $j \in I(\bar{L}, T)$ such that $psn_i = psn_j$. For each message, j, check if $t_j < t_d$, if so return to Step 6.

Step 9: Calculate the distance between i and d, $DIS(p_d, x_i)$.

Step 10: Calculate the estimated distance the vehicle traveled towards the destination from initial position, $DIS(p_d, x_D)$ using the following:

$$.5 * ((v_d * (t_i - t_d) + 0.5 * a_d * (t_i - t_d)^2) + (v_d * (t_i - t_d)))$$

Step 11: Calculate the error, $E(d, i)$ as follows:

$$ABS(DIS(p_d, x_i) - DIS(p_d, x_D))$$

If message i is a periodic, $mt_i = 3$, then add the following to the error, $E(d, i)$:

$$ABS(n - t_i)$$

Step 12: Select the PDM, i, with the lowest error, $E(d, i)$, add it to the destination list $D(b)$. Find all messages, $j \in I(\bar{L}, T)$ such that $psn_i = psn_j$ and append them to the destination list. Set $d = j_{MAX}$. If $I(L, [t_d, n])$is empty, r_D has been found, proceed to Step 14.

Step 13: If new message, d, is a periodic, $mt_d = 3$, or $t_d > n$ set n equal to the following:

If $v_d \leq LST, n = n + 4\ seconds$
Else if $v_d \geq HST, n = n + 20\ seconds$
Else $n = n + \left(4 + \frac{v_d - LST}{HST - LST} * 16\right)$

Return to Step 5.

Step 14: Set $o = b$. Repeat steps 15-23 until r_O has been found.

Step 15: For each link that could be travelled on from the initial PDM link, $L \in A(L_o)$, find all PDMs on those links at time between time t_o and , $I(L, [t_o, n])$.

Step 16: For each PDM, $i \in I(L, [t_o, n])$do steps 17-21.

Step 17: If x_i is downstream of p_o return to Step 17.

Step 18: Find all messages, $j \in I(\bar{L}, T)$ such that $psn_i = psn_j$. For each message, j, check if $t_j > t_d$, if so return to Step 17.

Step 19: Calculate the distance between i and , $DIS(p_o, x_i)$.

Step 20: Calculate the estimated distance the vehicle traveled towards the initial position from the new position, $DIS(x_i, x_D)$ using the following:

$$.5 * ((v_i * (t_i - t_o) + 0.5 * a_i * (t_i - t_o)^2) + (v_i * (t_i - t_o)))$$

Step 21: Calculate the error, $E(o, i)$ as follows:

$$ABS(DIS(p_o, x_i) - DIS(x_i, x_D))$$

If message i is a periodic, $mt_i = 3$, then add the following to the error, $E(d, i)$:

$$ABS(n - t_i)$$

Step 22: Select the PDM, i, with the lowest error, $E(o, i)$, add it to the origin list $O(b)$. Find all messages, $j \in I(\bar{L}, T)$ such that $psn_i = psn_j$ and append them to the origin list. Set $o = j_{MIN}$. If $i \in I(L, [t_o, n])$ is empty, r_O has been found, proceed to Step 24.

Step 23: If new message, d, is a periodic, $mt_d = 3$, or $t_d > n$ set n equal to the following:

If $v_d \leq LST, n = n + 4\ seconds$
Else if $v_d \geq HST, n = n + 20\ seconds$
Else $n = n + \left(4 + \frac{v_d - LST}{HST - LST} * 16\right)$

Return to Step 14.

Step 24: Merge the origin list, $O(b)$, and the destination list $D(b)$, to form the full re-identified route, $ID(b)$. Print $ID(b)$. Return to Step 1. If all initial PDMs have been re-identified, terminate.

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free “Help Line” 866-367-7487
www.its.dot.gov

FHWA-JPO-17-516

U.S. Department of Transportation