DEPARTMENT OF TRANSPORTATION

Framework and Guidelines for the Development of a Twin Cities Mesoscopic DTA Model

John Hourdos, Principal Investigator
Minnesota Traffic Observatory
Department of Civil, Environmental, and Geo- Engineering
University of Minnesota

April 2017

Research Project
Final Report 2017-10

To request this document in an alternative format, such as braille or large print, call 651-366-4718 or 1-800-657-3774 (Greater Minnesota) or email your request to ADArequest.dot@state.mn.us. Please request at least one week in advance.

Technical Report Documentation Page

1. Report No.
MN/RC 2017-10

2.

3. Recipients Accession No.

4. Title and Subtitle
Framework and Guidelines for the Development of a Twin Cities Mesoscopic DTA Model

5. Report Date
April 2017

6.

7. Author(s)
Derek Lehrke, John Hourdos

8. Performing Organization Report No.

9. Performing Organization Name and Address
Minnesota Traffic Observatory
Department of Civil, Environmental and Geo- Engineering
University of Minnesota
500 Pillsbury Dr SE
Minneapolis, MN 55455

10. Project/Task/Work Unit No.
CTS #2015010

11. Contract (C) or Grant (G) No.
(c) 99008 (wo) 153

12. Sponsoring Organization Name and Address
Minnesota Department of Transportation
Research Services & Library
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code

15. Supplementary Notes
http:// mndot.gov/research/reports/2017/201710.pdf

16. Abstract (Limit: 250 words)

Large-scale Mesoscopic traffic simulation is a newly adopted tool due to recent advancements in traffic modeling as well as computer hardware. New studies show that modeling on a scale necessary to answer complicated questions such as diversion patterns around multi-corridor work zones is feasible. As with many research projects, the original objective of this project was adjusted to maximize the benefit from the final product. The initial objective was to create a framework and guidelines for the development of a Twin Cites Mesoscopic Dynamic Traffic Assignment (DTA) model. Discoveries during the course of the project as well as MnDOT priorities and urgent needs directed the project away from the development of guidelines and more toward the proof-of-concept and the development of the foundation for such a metro-wide model. In addition, a parallel MnDOT project, undertaken by a consulting group using the DynusT application, developed an almost metro-wide model. The project described in this report, changed its scope to treat this parallel project as a case study and identify its future utility beyond its immediate goals, which were to determine the most cost-effective construction phasing for several projects during the 2017-2020 construction seasons.

17. Document Analysis/Descriptors
mesoscopic traffic flow, microscopic traffic flow, Dynamic Traffic Assignment, traffic simulation

18. Availability Statement
No restrictions. Document available from:
National Technical Information Services,
Alexandria, Virginia 22312

19. Security Class (this report)
Unclassified

20. Security Class (this page)
Unclassified

21. No. of Pages
163

22. Price

Framework and Guidelines for the Development of a Twin Cities Mesoscopic DTA Model

FINAL REPORT

Prepared by:

Derek Lehrke
John Hourdos
Minnesota Traffic Observatory
Department of Civil, Environmental and Geo- Engineering
University of Minnesota

April 2017

Published by:

Minnesota Department of Transportation
Research Services & Library
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

This report represents the results of research conducted by the authors and does not necessarily represent the views or policies of the Minnesota Department of Transportation or the University of Minnesota. This report does not contain a standard or specified technique.

The authors, the Minnesota Department of Transportation, and University of Minnesota do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to this report because they are considered essential to this report.

ACKNOWLEDGMENTS

We would like to thank the Minnesota Department of Transportation for supporting this project. We would like to acknowledge the help, support, and cooperation of Jim Henricksen with MnDOT, Jonathan Ehrlich and Mark Filipi of the Metropolitan Council, the Technical Advisory Panel for its input and direction in addition to all the local consulting firms that were interviewed as part of this project. Special thanks is also required for Daniel Morgan, from Caliper, for donating his time and his team to construct a rough model as part of the final task in this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AADT	Average Annual Daily Traffic
ABM	Activity Based Model
ATDMS	Advanced Traffic and Demand Management
ATR	Automatic Traffic Recorder
API	Application Programming Interface
CTSCS	Centralized Traffic Signal Control Software
DTA	Dynamic Traffic Assignment
FHWA	Federal Highway Administration
GIS	Geographic Information System
GUI	Graphical User Interface
HCM	Highway Capacity Manual
HOV	High Occupancy Vehicle
ITS	Intelligent Transportation Systems
Macro	Macroscopic
Meso	Mesoscopic
Meso-DTA	Mesoscopic Dynamic Traffic Assignment
Met Council	Metropolitan Council
Micro	Microscopic
MnDOT	Minnesota Department of Transportation
MTO	Minnesota Traffic Observatory
OD	Origin/Destination
ODME	Origin Destination Matrix Estimation
RFP	Request for Proposal
RTDFM	Regional Travel Demand Forecasting Model
SOV	Single Occupancy Vehicle
SQL	Structured Query Language
STA	Static Traffic Assignment
TAP	Technical Advisory Panel
TAZ	Traffic Analysis Zone
TBI	Travel Behavior Inventory
TDM	Travel Demand Model
TMC	Turning Movement Counts
TRB	Transportation Research Board
UTDF	Universal Traffic Data Format
VMT	Vehicle Miles Traveled

EXECUTIVE SUMMARY

Large-scale Mesoscopic traffic simulation is a newly adopted tool due to recent advancements in traffic modeling as well as computer hardware. New studies show that modeling on a scale necessary to answer complicated questions such as diversion patterns around multi-corridor work zones is feasible. As with many research projects, the original objective of this project was adjusted to maximize the benefit from the final product. The initial objective was to create a framework and guidelines for the development of a Twin Cites Mesoscopic Dynamic Traffic Assignment (DTA) model. Discoveries during the course of the project as well as MnDOT priorities and urgent needs directed the project away from the development of guidelines and more toward the proof-of-concept and the development of the foundation for such a metro-wide model. In addition, a parallel MnDOT project, undertaken by a consulting group using the DynusT application, developed an almost metro-wide model. The project described in this report, changed its scope to treat this parallel project as a case study and identify its future utility beyond its immediate goals, which were to determine the most cost-effective construction phasing for several projects during the 2017-2020 construction seasons.

The first goal in this effort was to determine if other large-scale models exist and if it was feasible to run those models on normal workstation computers. The team found multiple models that fit this description and in some exceptional cases, the networks were even modeled at the Microscopic level, which is computationally more intense and has greater data requirements than a Mesoscopic model. In this report, four modeling efforts, each adopting a different simulation application, were scrutinized. These examples were used to determine some of the basic data requirements as well as to outline significant drawbacks and benefits in the process of developing a large DTA traffic simulation model. Having enriched our understanding of the different aspects of the problem, a survey was designed and carried out through interviews with local stakeholders. The interviews, in conjunction with the knowledge from the previously mentioned models, allowed for a more targeted determination of the required data, desired outputs, and practical roadblocks and limitations in developing a Twin Cities model.

One important product of this effort, described in Chapter 6 of this report, is the development of a “compatibility matrix” for five commercially available mesoscopic simulation suites. This matrix identifies 40 components or features that are necessary for the development of an accurate model or for allowing the use of said model in studying fundamental transportation problems. For the five most widely used traffic simulation applications, an analysis of how each of these 40 components are handled, abstracted, or estimated was carried through. This effort augmented the produced results beyond the Yes or No indication for a certain component application pair. While all the researched software packages could potentially preform a sufficient DTA simulation of the Twin Cities, two stood out: Aimsun and TransModeler. Both of these applications are not only feature rich but also are able to perform multiple levels of simulation within the same model. This type of feature is extremely useful since it allows the user to perform large-scale mesoscopic simulations on the network while also being able to use the same geometry to perform microscopic analysis of specific subareas. Alternatively, the

most readily implementable DTA mesoscopic simulator would most likely be Cube Avenue since the regional network, on the Travel Demand Modeling (TDM) level, already exists in Cube Voyager. Based on how Cube Avenue works, such a mesoscopic DTA model would only be useful in studies of major changes in capacity as it would not be able to examine the effects of more complex scenarios involving weaving, complex traffic control, or other lane-dependent traffic components.

As earlier studies have also shown, there is a lot of experience in the local transportation community on the use of TDM's in terms of calibration, maintenance, and their the general use of them. However, TDMs often incorporate a very large geographical area that is computationally difficult and nearly impossible to calibrate without considerable effort. Furthermore, the primary objective of most TDMs is to forecast traffic demand throughout the entire geographical region based on changes in land use and demographics and not to analyze the effects of such changes on traffic at the operational, road segment, level. The same can be said for microscopic models. Both of these model types are firmly established in the transportation modeling community with a lot of experience and past projects to learn from. The newest type of large-area mesoscopic simulation or even large microscopic simulation models do not have well-established steps for constructing such a model. This report includes a chapter that specifically outlines the essential data needed for such modeling efforts and potential issues in acquiring them as well as general suggestions for model development and output. Topics include Aerial Imagery, Signal Timings, Freeway and arterial traffic counts/speeds, congestion locations, saturation flow, and Origin-Destination data.

The final effort described in this report worked toward laying the foundation for a Twin Cities area model based on the TransModeler application. As noted earlier, two simulation applications, Aimsun and TransModeler, were identified as having all the required features at the required accuracy and realism to cover the needs of a Meso-DTA user as well as being able to seamlessly transition to microscopic when necessary. In the case of AIMSUN, the research team had extensive experience and had already developed a Twin Cities wide hybrid DTA model as part of an earlier project. During that project, it was concluded that the current execution speed of such a model was too slow for wide use and adoption by MnDOT. TransModeler has shown to possess impressive performance in terms of speed and ability in modeling large geographic regions. In addition, Caliper, the software developer, offered to assist the research team in testing some of the software features under network size realistic conditions. The firm's help was offered specifically in building a large network geometry, importing a sample of MnDOT's arterial traffic control information, and most importantly importing and adjusting the network demand information as received from the Metropolitan Council's Regional Travel Demand Forecasting Model. While it would have been better if the demand had come from the new Activity Based Regional Planning Model, the integration procedure would have been considerably more complex due to the increased level of detail and would have hindered the project's timeline. Nevertheless, the older TDM provided a seed demand matrix that the Caliper Dynamic-ODME procedure was able to use and produce a more refined demand based on data collected by freeway detectors. During this proof-of-concept, reasonable benchmarks in terms of effort were collected as well as run times for each step. The model performed reasonably well, although very close to the limits specified by the stakeholders interviewed.

Based on the findings of this report, the groundwork has been completed and the procedures are laid out for MnDOT and other entities to pursue the creation of a calibrated Twin Cities DTA model in TransModeler or Aimsun. For TransModeler, Caliper and the research team have delivered a rough network with limited resources that is showing great potential. In Aimsun, a more detailed and semi-calibrated model exists, albeit running too slow for active use right now. This can change as computational power increases and software upgrades improve the efficiency of the model. By building the model in either software suite there is also room for expansion and additional detail that cannot be done in the current DynusT-based DTA model without moving the network to a different software package.

CHAPTER 1: INTRODUCTION

The economic impact of traffic management grows each day. Well-designed and well-managed roadway systems reduce the cost of transporting goods, cut energy consumption, and save countless person-hours of driving time. More and more, transportation system operators are seeing the benefits of strengthening links between planning and operations. As new complicated Advanced Traffic and Demand Management Systems (ATDMS) are implemented, the more evident it gets that their influence reaches beyond the individual corridor they were designed for. A critical element in improving transportation decision-making and the effectiveness of transportation systems related to operations and planning is the use of analysis tools such as traffic simulation models.

The federal government through the, now thirteen volumes long, Traffic Analysis Toolbox has illustrated the need and benefits from more sophisticated modeling traffic analysis tools. The latest volume "Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling" stresses the importance of considering route choice in any project analysis. As the urban network increases in density the effects of incidents, construction zones, transportation demand management strategies, Integrated Corridor Management (ICM) strategies, Intelligent Transportation Systems (ITS), as well as capacity increasing strategies gets increasingly more difficult to understand and plan for. In the Twin Cities we have already realized the need for a Regional Travel Demand Model (RTDM) and there is no capacity increasing project that is not planned without the help of a microscopic traffic modeling. These two are at the two ends of the traffic simulation spectrum, macroscopic and microscopic (Micro). The macroscopic considers route choice but it understands only static traffic conditions. The microscopic emulates real world conditions but would require a room sized computer to consider inter-corridor or regional effects, assuming you can find enough data and labor to calibrate it. MnDOT has realized the need for a bridge between the two levels. This bridge is Mesoscopic traffic simulation with Dynamic Traffic Assignment (Meso-DTA). There is also Micro with DTA but it is currently technologically challenging and requires a significant cost in order to calibrate it. Mesoscopic simulation and DTA are new concepts and there are several commercial applications that claim to offer such functionality. Unfortunately, there are no two of them that follow the same modeling methodology.

This report is a summary of the work done to create a Framework and best practice for the implementation of a Mesoscopic DTA model of the Twin Cities region. Due to the disjoint between research and construction, funding this project was not able to be completed before MnDOT needed to have a simulation model ready to determine the most cost effect construction phasing for several projects in the 2017-2020 construction seasons. A Mesoscopic DTA model was assembled to accomplish this task before the recommendations that this report was to provide could be completed and therefore the scope of this project was altered towards its completion.

It begins with a breakdown of important terms and fundamental definitions that need to be defined to inform the reader of assumptions and definitions used by the research team. The report then moves into a brief description of 4 case models already developed in the United States with detailed accounts found in Appendix A. Following this are sections that outline the needs and requirements of a Meso-DTA model as derived through interviews with local stakeholders and comparing them to the capabilities of commercially available software. All of this information was synthesized into recommendations of required data and a comparison of the Pro/Cons of a large regional model vs smaller project-size models.

It was during the last two tasks that an opportunity presented itself, because of the previously mentioned Meso-DTA model, to devote the remaining project time to evaluating the model built for MnDOT or to evaluate the effort required to build a similar model in a different software package. After presenting the options to the Technical Advisory Panel (TAP) they voted for the latter and thus the remaining portion of the report is dedicated to evaluating the effort needed to establish a DTA model of the Twin Cities in TransModeler.

CHAPTER 2: TRAFFIC SIMULATION METHODS

While this project was to develop the guidelines for a mesoscopic simulation model it is necessary to understand where mesoscopic simulation lies in current state of traffic modelling practice while also understanding the limitations and advantages of other methods. There exists a grey area in understanding the separation between the Static Traffic Assignment (STA) step in TDMs and the term Macroscopic simulation. In reality, traffic assignment, static or dynamic, is not related, as a concept, with traffic simulation, but because many traffic simulation applications package a static traffic assignment module along with their microscopic and/or mesoscopic traffic simulation models a lot of people confuse it as a higher (macro) level of traffic simulation. Indeed, some of the most known simulation applications refer to the static traffic assignment module as Macroscopic Simulation. This is not only misleading but also inaccurate since Macroscopic Traffic Simulation is a higher level of traffic simulation that includes models that treat traffic as a compressible medium (a liquid) and utilize amongst others hydrodynamic relationships and the fundamental diagram to simulate flow conditions in links. Such models are still used in some cases because they are fast and for simple geometries provide reasonable results. Such models include Freeval, Kronos, and others.

2.1 TRAVEL DEMAND MODELS

As is the current state of practice, high-resolution traffic simulation models are often not built into Travel Demand Models (TDM). Therefore, many traffic simulation models use a regional travel demand model to generate the demand necessary for them to run. TDM’s combine local socioeconomic data as well as responses from distributed surveys to predict how people travel to and from points of interest. The two most common forms of travel demand models are the Four-Step or trip based model, which some industry professionals would refer to as a Macroscopic TDM, and the newer Activity-Based Models (ABM) which are sometimes refer to as Microscopic TDM’s. Modeling suites that can run these types of TDM’s included Voyager (CUBE), Emme/2 (Inro), TransCAD (Caliper), Visum (PTV Vision), and Aimsun (TSS).

2.1.1 Macroscopic Travel Demand Modeling (Four-Step)

As is deployed by many State, Regional, and Municipal agencies the four-step model is a trip based model that includes four primary steps. The first step, Trip Generation, estimates the number of trips to be generated and attracted in each Traffic Analysis Zones (TAZ) in the region being modeled. The second step, Trip Distribution, connects the trips generated in step one to form an origin and destination for each trip. The third step, Mode Choice, determines how each of the trips generated will travel, whether it be by car, carpool, transit, or any other method programmed in. The fourth step, Traffic Assignment, determines the specific route from which each trip found in step two, using the mode selected in step three. The four steps are repeated in a loop until no further change in the O/D matrices is observed and the volume at each link changes only marginally.

2.1.2 Microscopic Travel Demand Modeling (Activity Based)

Activity-Based models represent the next stage of regional travel demand forecasting. While they are not yet as standardized as the four step models they are getting more popular in their implementation as they incorporate significant advancements that planners and forecasters need. These advancements are made possible because at its core the models work at a disaggregated person level rather than at the TAZ level as is the case with most 4-step/trip based models. ABM's represent each of these person's activities and travel choices over the course of an entire day to determine and prioritize their activities (work, shopping, school, etc.). That person's schedule is then filled with activities until the time available to participate in additional activities diminishes.

Activity Based Models involve additional steps not included in four step models due to the added complexity of disaggregating out to the person level. However, like the four step models, ABM's can be derived into four core steps. Each of these steps can also vary greatly in terms of how many different models are in each one, how they are implemented, and in what order. These types of differences are covered extensively in the Activity Based Modeling Primer[1] published under SHRP2 in 2014.

- Synthetic Population Synthesizer – Generates the "persons" in the modeled region.
- Long-Term and Mobility Choice – Used to determine long term effects of travel behavior and what modes of transportation are available to a given person. Examples include whether or not to own a car, where to work, whether to buy a transit pass, etc.
- Daily Activity Pattern and Tour trip detail – Generates the tours that each person will take
- Trip assignment – Assign the tours to the network

2.2 STATIC TRAFFIC ASSIGNMENT

Static traffic assignment uses a simplified abstraction of the traffic simulation model geometry. This level treats the roadway network only as links and nodes also referred to as a "stick network". Using Volume Delay Functions (VDF's), or an equivalent measure, they model the impact of link volume on travel time and in extent route choice. The inclusion of static traffic assignment into a traffic simulation application is necessary for practical and model related reasons. Practically, what the traffic modeler receives as input information is a set of O/D matrices describing the demand, means to build the network geometry (a GIS representation of the road network used by the TDM, digital maps, or aerial imagery), as well as information regarding traffic control and traffic measurements. It is good practice to repeat the traffic assignment performed by the TDM in the new platform and catch mistakes or wrong abstractions in the

[1] http://www.trb.org/Main/Blurbs/170963.aspx

geometry between the two applications. Although in the surface two geometric representations may look the same the underlying assumptions used in the model can vary a lot.

The model related reason involves the paths associated with each O/D pair. Although the TDM has produced sets of paths, and their volume ratios, for each O/D pair this information rarely is passed on. Regardless if a microscopic or mesoscopic traffic model is to be used, results and performance is greatly enhanced if these paths are provided as a starting point to the simulation. One can think the paths produced by the static traffic assignment as the historical paths people formulate over time as they experience the road network. Without these paths one can consider the allegory of all simulated drivers being tourists, having no knowledge of the road network, guided only by the conditions currently experienced. Regardless if later the simulation is paired with a form of dynamic traffic assignment, providing the historical paths improves performance and accuracy. More discussion on this is included in a later section.

Conceptually, in the TDM process, instead of using static traffic assignment, the simulation model can be used to estimate link travel times. Transims, a federally supported TDM and simulation application employs such a combination but it requires a supercomputer in order to run. Simpler ways of integrating TDM's and simulations can be achieved by partially replacing the static traffic assignment with a simulator employing DTA and produce the same level of accuracy with fewer resources. Such an integration was the subject of a recently concluded project ("Evaluating Twin Cities Transitways Performance and their Interaction with Traffic on Neighboring Major Roads") that replaced the traffic assignment portion of the Twin Cities RTDFM, with a hybrid traffic simulation model.

2.3 MICROSCOPIC TRAFFIC SIMULATION

Microscopic models are high-resolution with great detail in terms of geometry and control. Car following, lane changing, and other driving behavior models are used within a fixed-time-step or event based framework as vehicles propagate through the network making decisions which cumulatively produce traffic patterns. Vehicle behavior is governed by parameters such as reaction time, acceleration and deceleration rates, gap acceptance, and so forth. It is also required to have accurate geometry, signal timings, and other control information so that the effects of queuing at intersections, congestion buildup, weaving, and all other dynamic elements of traffic can be captured.

The limits of microscopic simulation are the computational requirements, the data requirements for calibration/validation, and the effort needed to build the model. Each traffic signal, intersection turning movement, and roadway link must be calibrated to mirror behavior in the real world or otherwise the network could be anywhere from gridlocked to completely clear. All of these calibration efforts require significant quantities of data (traffic counts, travel time runs, spot speed studies, etc.) in order to calibrate the model to match these real-world effects.

Micro models can employ all sorts of methods for assigning vehicles into paths and guiding them as they traverse the network. From STA, DTA, or even fixed routes can be used. More discussion on this part follows on a later section.

2.4 MESOSCOPIC TRAFFIC SIMULATION MODELING

While "Mesoscopic" simulation is not yet a fully defined method it does represent a middle ground between macroscopic and microscopic models. One popular meso methodology, and maybe the best example of the mix between macro and micro involves the simulation of individual vehicles that have an origin and a destination as well as some driving behavior parameters (micro) but their movement inside each roadway link is defined by aggregated traffic flow relationships (macro). Since each software vendor incorporates and defines mesoscopic simulation differently, as well as updating their programs and expanding the features of what mesoscopic simulation can do, a general overview of mesoscopic simulation is described in the following paragraphs.

TDM models can only provide so much detail, as described in the previous section, in simulating traffic conditions. In many cases, projects require more in-depth simulation results and require features not available in TDM models. One major difficulty present in most TDM models is that they miss the dynamic nature of traffic and in extend cannot do DTA well or not at all. Better clarified in a later section, to perform DTA it is best that the path selection is performed by individual vehicles or small groups of vehicles. Although DTA can always be used along with microsimulation, the computing resources required reduce the feasible size of the network modeled. This is where mesoscopic models come into the picture. These models have the ability to model large study areas that would be computationally infeasible for a microsimulation model. They are thus able to provide users with more detailed information than macroscopic models while still feasible to build.

Generally mesoscopic models use individual vehicles or cells/packets of vehicles to model traffic and put these vehicles onto geometrically correct road networks or links. This added level of detail allows the simulator to consider real world traffic conditions such as the following.

- Weaving - While some of the software suites being tested are able to simulate vehicles down to the lane level, such as Aimsun, Dynameq, and TransModeler, others such as DynusT and Cube Avenue use a link based network so they cannot replicate the congestion caused by weaving between lanes. This is an important feature to consider if weaving sections are common in the network so that congestion can form more naturally. Still one should consider the many ways weaving can be modeled; most mesoscopic models do not directly model the effect of lane changes (micro) but more the macro effect of lane flows mixing.
- Lane Utilization – As with weaving, lane utilization can be very important in some networks that suffer from congestion due to single lane breakdowns or turn pocket overflows. For example, a local single lane breakdown in Minneapolis occurs at I-94 West approaching the Lowery Hill tunnel. At this location the right most lane breaks down during rush-hour due to lane changes

being made further downstream while the remaining two left lanes will often have speeds that on average are 15+ mph higher than the rightmost lane. In simulation models that do not have lane level detail this type of breakdown cannot be replicated accurately and thus assumptions must be made to approximate the single lane breakdown over the entire three lane section

- Queueing – As is common in all major cities congestion will cause queues that often expand farther than any one section/link in a simulation model. Mesoscopic model allow queues to propagate through the network and into down-stream sections/links. Queuing can happen on a lane by lane level or at a link level.
- Traffic Signals – Almost every static traffic assignment does not have any way to replicate delay caused by signals. While some average delay could be applied to every node to help replicate the effect of the signal on traffic it does not account for signal coordination or other advance signal properties. These types of advance signal properties can be modeled in mesoscopic simulators although their implementation appears to vary among the different software and will be an important feature to explore.

One of the key differences in mesoscopic simulation that differentiates it from microscopic is how they handle traffic flow. Since every software suite is different it is hard to generalize on how mesoscopic traffic flow is summarized. A common method is to use a fundamental flow-density diagram approximated into a simplified triangular form. This diagram as shown in Figure 1 approximates the saturation flow rate of a section given its density and where the static parameters of Free-Flow Speed (FFS), Effective Vehicle Length (EL), Jam Density (K_j) and Reaction/Response Time (RT) are defined at the beginning of the simulation.

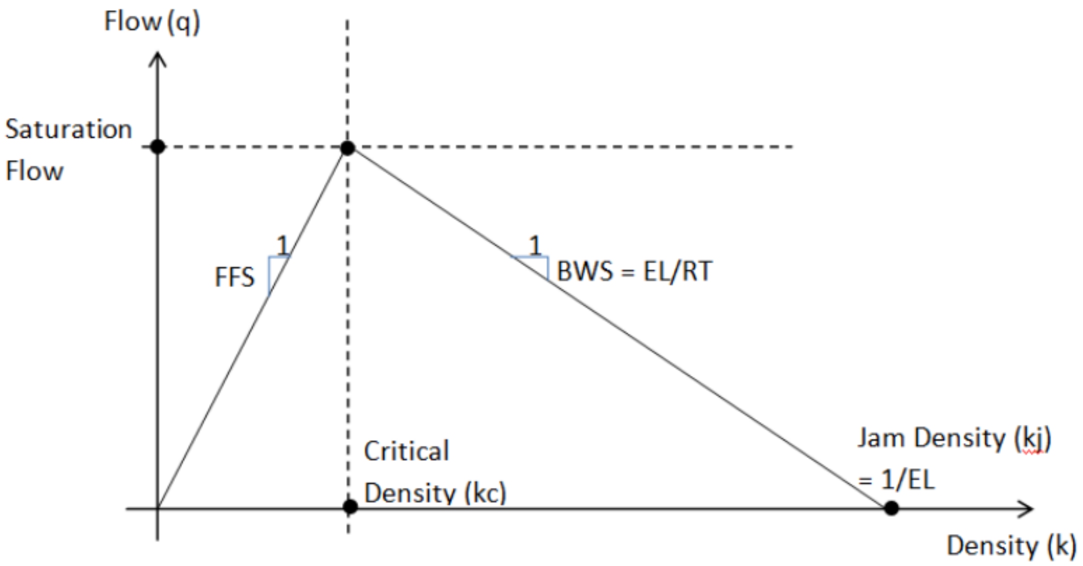


Figure 1: Flow-Density diagram approximated into a simplified triangle form

Based on research already conducted by the Minnesota Traffic Observatory (MTO), a complete understanding of how each software represents and models these key functions is critical to using mesoscopic simulation. For example, Aimsun incorporates a similar model and it was found that the Jam Density and FFS had very little effect, if any, on the assigned flow of a section. However, a small change in reaction time (as small as 0.1 seconds) could greatly alter the flow of a section and change the saturation flow rate by several hundred vehicles per hour. These types of correlations between different variables are essential for calibrating a mesoscopic network.

2.5 HYBRID SIMULATION

While microscopic simulations provide the highest resolution of data, they are memory and computationally intensive. Mesoscopic simulations require less resources to complete, but do not offer the same level of detail. Using a high-resolution, microscopic simulation core surrounded by a lower-resolution, mesoscopic outer layer, regions of particular interest can be examined fully while reducing the total computation expense. Additionally, including a mesoscopic layer allows vehicles to find routes outside of the microscopic region if congestion or incidents cause significant disruptions in the microscopic areas (see Figure 2, (TSS Aimsun manual, pg442)).

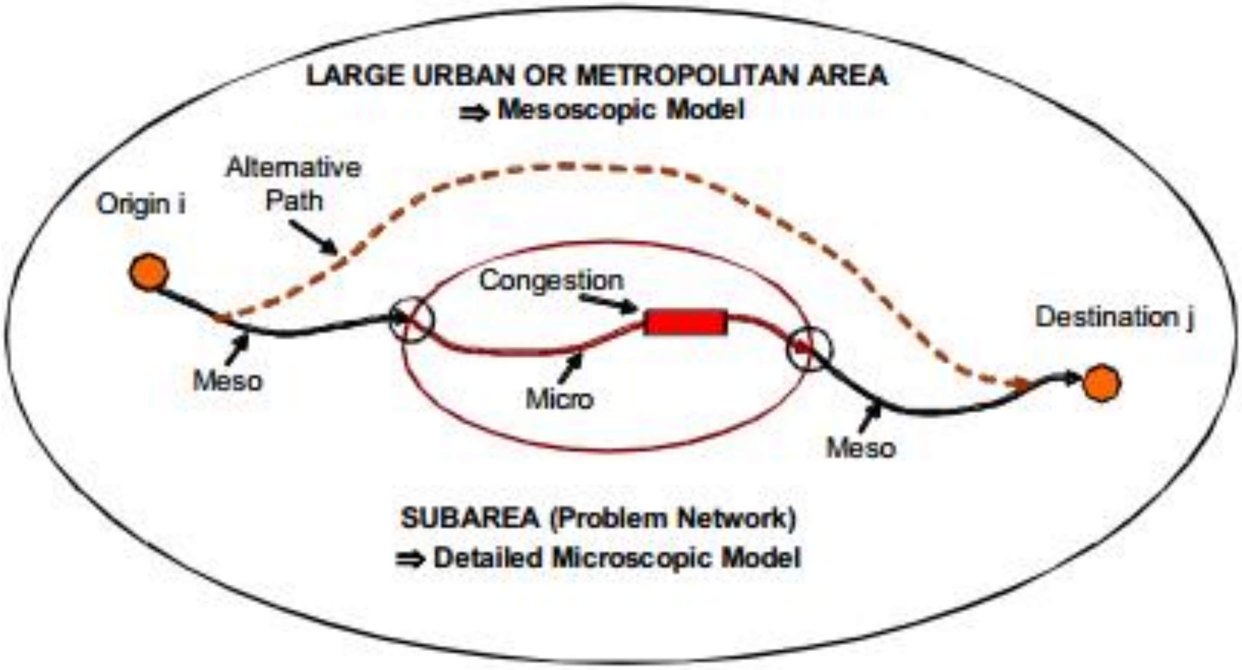

LARGE URBAN OR METROPOLITAN AREA
⇒ Mesoscopic Model
Alternative Path
Origin i
Congestion
Destination j
Meso
Micro
Meso
SUBAREA (Problem Network)
⇒ Detailed Microscopic Model

Figure 2: Hybrid modeling to locate alternative paths.

Within the focus area (microscopic simulation), the primary path between the origin and destination develops congestion. If only the microscopic region is considered, no alternative is available and vehicles must wait through the congestion with large delays. However, if the mesoscopic region is also included, vehicles moving from origin “i" to destination “j” can select alternate routes, which do not utilize the affected links. In this way, a more accurate representation of driver decision making can be modeled and the high level of detail is maintained within the corridors of interest, but computing needs are dramatically reduced (compared to simulating the entire area of interest microscopically).

CHAPTER 3: ASSIGNMENT TYPES

As was already eluded there are two types of traffic assignment, static traffic assignment (STA) and dynamic traffic assignment (DTA). Although, DTA is usually considered part of simulation, conceptually it is an isolated component only necessary if demand is given in the form of time dependent O/D matrices and vehicle movement is governed by paths between O/Ds. DTA or any kind of assignment is not necessary for either micro or meso simulation. Both of these can function by static or time dependent entry volumes and turning percentages at each node. This non-DTA abstraction is generally preferred if the roadway does not offer multiple routes from entry to exit i.e. modeling a freeway section without including any arterial streets. In more complex cases, the volume/turning percentages method of describing demand rarely produces realistic results hence a method for selecting between possible paths is necessary. There are three distinct types of DTA that can be implemented as described in Chapter 2 of a “Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling” a FHWA publication. These methods include the Dynamic User Equilibrium (DUE), Dynamic System Optimal (DSO), and finally the Non-Iterative or One-Shot DTA.

3.1 DYNAMIC USER EQUILIBRIUM

Dynamic traffic assignment models are most often assigned and set to converge based on a formulation by Ran and Boyce (3):

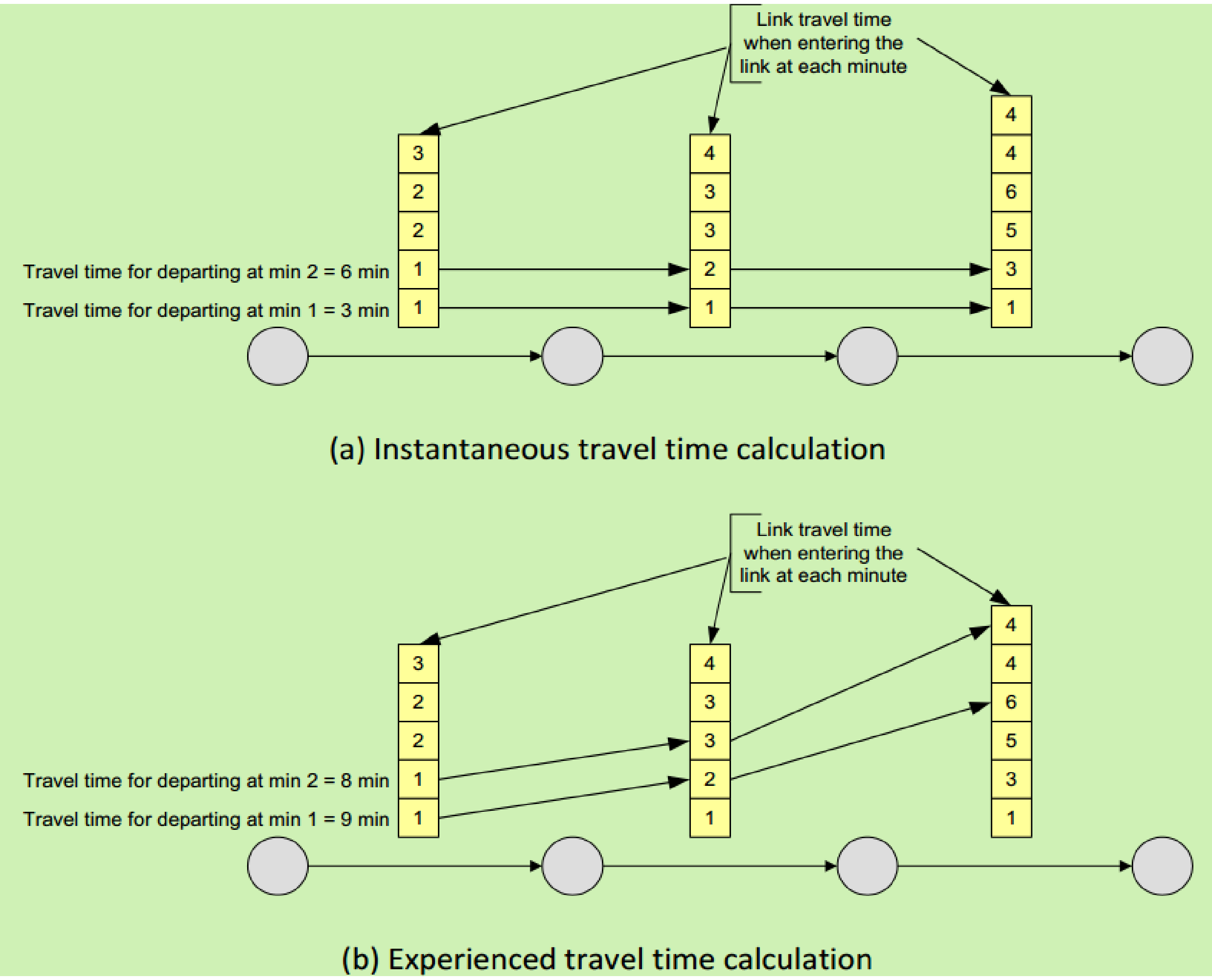
> *“If, for each OD pair at each instant of time, the actual travel times experienced by travelers departing at the same time are equal and minimal, the dynamic traffic flow over the network is in a travel-time-based dynamic user equilibrium (DUE) state.”*

As vehicles are loaded into the network, they select paths based on their “cost or utility” which is usually interpreted as the shortest travel time but can be a function of other parameters that describe the “cost” to facilitate other considerations such as toll roads. This procedure is repeated N times, terminating once some upper limit on iterations has been reached, or when the DUE has reached a significant level of convergence. This second criterion is usually based on a statistic called the Relative Gap or R-Gap. The R-Gap, proposed by Janson (4), is the ratio of ‘excess delay’ experienced by all users as compared to their possible minimum paths. It is designed to give a percentage of the total gap, the difference between an OD’s current route and the shortest route, divided by the total shortest path times. As the R-Gap approaches zero it is to be inferred that no traveler between any OD, during any departure interval, was able to find a shorter route.

3.1.1 Instantaneous Vs. Experienced Travel Times

Under the DUE assignment, travel times of travelers are used to calculate their shortest paths. These travel times are different based on whether experienced or instantaneous travel times are considered. Instantaneous travel times use the current travel times of all links when the vehicle enters to determine the shortest path. This however does not account for the fact that travel times can change dynamically through the simulation, either due to congestion, an active traffic management system, or other factors that could impact travel time. This instantaneous look at the system could result in many vehicles choosing routes which converge into a bottleneck which was not active at the time they entered, resulting in significantly greater travel times than alternative shortest paths.

Experienced travel times use previous iterations to determine the travel time of a vehicle using the travel time of every section in its path when it would reach it. In other words, it looks into the “future” to see what the travel time on every section will be, given the time it takes to get there. This would be the equivalent of a commuter who knows that if they do not leave before a certain time they will hit congestion based on past experiences. This is more appropriate for determining equilibrium, but usually is more computationally intensive. Figure 3 (DTA Primer, pg.15) depicts a sample calculation on an arbitrary network of the differences between these two types of travel time (1).


Link travel time when entering the link at each minute
3
2
2
1
1
4
3
3
2
1
4
4
6
5
3
1
Travel time for departing at min 2 = 6 min
Travel time for departing at min 1 = 3 min

(a) Instantaneous travel time calculation

Link travel time when entering the link at each minute
3
2
2
1
1
4
3
3
2
1
4
4
6
5
3
1
Travel time for departing at min 2 = 8 min
Travel time for departing at min 1 = 9 min

(b) Experienced travel time calculation

Figure 3: Experienced vs. instantaneous travel time

3.2 DYNAMIC SYSTEM OPTIMAL

Another implementation is the dynamic system optimal solution. This is very similar to the DUE solution but instead of traveler's travel time being minimized the total "cost" of the system is minimized even if every traveler's travel time is not. This type of assignment assumes that there is a centralized management system that forces individual travelers to routes that minimize the collective travel time. This is infeasible in almost all daily travel cases since road users generally act independently and tend to minimize their own travel time rather than coordinate to improve overall travel times at their own expense. It can however be used to analyze evacuation strategies or other cases where road users can't act independently.

3.3 ONE-SHOT DYNAMIC TRAFFIC ASSIGNMENT

While equilibrium aims to emulate the long-term selection of routes by travelers in a network, it does not cover all scenarios. Equilibrium assumes that all travelers have some knowledge of recurring congestion and make reasonable choices to reduce their travel time. One-Shot DTA is meant to model the effects of scenarios such as short-term work zones, incidents in critical sections, and items such as tourist attractions or others not familiar with the network. In most cases, One-shot DTA is assisted by the provision of historical routes as described earlier or even the routes produced as the result of a DUE. In such cases, some percentage of the vehicles force the predefined routes either static (historical) or time dependent (DUE), some are assigned new routes based on current conditions and follow them to the end, while in many cases a third group is allowed to re-route during their trip given updated information regarding traffic conditions. In complex networks, DUE simulations can take very long times and therefore not efficient to be repeated often while One-Shot are faster and if performed properly can produce credible results.

CHAPTER 4: CASE STUDIES

Given that many agencies have come to the same conclusions that TDM's cannot solve all questions, and turn to higher resolution models to answer them, many case examples were mentioned however few had reports readily available that described them. As can be seen in Appendix A, which outlines just a few of the larger models that had reports available, the scale at which DTA can be achieved is ever increasing. However, as was noted in this report and as described in the papers that outline these models, there are many requirements and challenges to building a functional DTA model. These types of models are a lot more data intensive in terms of calibration and validation than their older counterparts. All of the developers of the mentioned models derived ways to mitigate these challenges and produce DTA models that could answer complex questions that their previous models could not. It is the research team's opinion that, given the success of these models, a DTA model is not only feasible for the Twin Cities but could be seen as a necessary leap that must be done in order to answer the complex questions sought by the users of the Twin Cities network. MnDOT has already undertaken one such attempt with the creation of a Construction Phasing Model that discussed in detail as part of section 9.1.

A comparison of the different models is presented in **Error! Reference source not found.** and the major t akeaways from the models that will prove useful in the implementation of a Meso-DTA model are outlined below.

- Manhattan Traffic Model
    - Signals – The entire process of having to have 4 different ways to program signals could have been avoided if a centralized database of signals had been available. By using 4 different methods to determine control plans for the MTM, badly programmed or incorrect signal timings could have been introduced leading to calibration issues later in development.
- San Francisco DTA Model
    - Model development – The San Francisco model appeared to have one of the most useful frame work for developing the model by generating the main model from scratch each time. Since many DTA models will need to be developed in short periods of time the ability to have multiple people working on it is essential.
- Detroit DTA Model
    - Calibration data - The integration of data from traffic.com (now HERE) is an important step since data is often not available for arterial streets in the network. This data is essential in combination with freeway data to form a complete picture of the network for the purposes of calibration.
- Jacksonville DTA model
    - Integration with CUBE – The Jacksonville model used a direct integration with CUBE and thus represents that the software suite could potentially integrate with the RTDFM easily and closely resembles the frame work that a Twin Cities DTA model would follow.

Table 1: Four case model comparison table	JACKSONVILLE DTA MODEL	DETROIT DTA MODEL	SAN FRANCISCO DTA MODEL	MANHATTAN TRAFFIC MODEL
ROAD LENGTH	5,440 miles	NA	NA	1,627 km
LANE LENGTH	11,870 miles	NA	NA	4,520 km
ZONES (TAZ)	1,862 zones in the trip-based model (Trucks), and 492,684 parcels (Cars)	~1200 TAZ	976 TAZs, 22 external stations	1,583 centroids
NUMBER OF SECTIONS	42,000 links and 31,000 centroid connectors	~20,000 (links)	37,000	9,720
NUMBER OF INTERSECTIONS	There are 34,320 nodes, 1,300 of which are signalized intersections, and 14,508 of which are boundary nodes	NA	1,115 signalized, 3,726 stop controlled (15,000 nodes)	3,566
SIMULATION DURATION	AM Peak 4 hr (5:00 – 9:00 AM) MD Peak 6.5 hr (9:00 AM – 3:30 PM) PM Peak 3 hr (3:30 – 6:30 PM).	AM Peak 4hr (6:00-10:00) PM Peak 5hr (2:00-7:00)	2:30 to 7:30 (5hr), including 1hr warmup/cool down	from 5:30 to 10:00 (4h 30 min)
TOTAL DEMAND	AM Peak ~ 738,000 MD Peak ~ 1,450,000 PM Peak ~ 866,000	NA	535,200 cars, 84,200 trucks	1,437,542 vehicles
TIME TO SIMULATE	NA	NA	52.5 hours	11 minutes (Time to complete 1 iteration as part of a DUE)

VALIDATION CRITERION	%RMSE at 555 count locations	15-minute traffic counts from traffic.com (# of locations NA)	Caltrans Travel Forecasting Guidelines	±15% difference between measured and simulated flows
CLIENT	North Florida Transportation Planning Organization (NFTPO) and HNTB	Michigan Department of Transportation (MDOT)	San Francisco County Transportation Authority	NYC DOT
DATE OF COMPLETION	2010	2013	2012	2011
COST	NA[2]	NA	NA	NA

[2] Attempts were made to acquire the cost (both in time and money) but were not able to be acquired for this paper due to the reluctance of private companies and DOT's to disclose or inability to locate such statistics.

CHAPTER 5: LOCAL INPUT/STAKEHOLDER OPINION

This section is meant to represent a summary of the needs of the local stakeholders as well as the needs identified in other projects. To identify the needs and requirements the interviewee responses, the Minnesota Traffic Observatory's (MTO) experiences, and the reports of completed/ongoing Meso-DTA models throughout the country were used. A direct interview with at least one person from each of the major consulting firms/organizations was attempted but in some cases indirect interviews via Email, RFP's, TRB papers, and published/unpublished reports were used instead.

5.1 INTERVIEWEE RESPONSES

When questioned about what Meso-DTA meant to them most of the interviewees responded by stating that while they have heard of mesoscopic simulation and DTA, many do not differentiate between the two. It appears that nearly all of them, depending on the type of analysis, whether it be corridor or subareas, assumed that DTA was implied when using microscopic simulation. However most saw a difference in the fact that, while in concept, mesoscopic simulation may be able to be differentiated from DTA it would not be useful. This reasoning stems from the fact that if they were putting the effort into creating a network for mesoscopic simulation is was to achieve some greater level of detail and the ability for vehicles in the model to alter their behavior based on the networks conditions. The only case that arose in these interviews in which that did not apply was in terms of corridor analysis were alternative routes were not necessarily available for vehicles to choose.

While discussing which simulation software packages were used by the local consulting firms and MnDOT it became apparent that most only had experience in travel demand and microscopic modeling. Only a couple defined themselves as using mesoscopic modeling and of those all of them used PTV Visum or Cube Avenue as their mesoscopic simulator of choice. The most popular option for microsimulation was CORSIM (originally developed by FHWA) for freeway modeling, Vissim when additional detail was necessary, and Synchro for traffic signal analysis. FREEVAL which is part of the Highway Capacity Manual (HCM) is also used to model freeway sections quickly but its functionality is limited. Cube Voyager was also mentioned since it is used to run the Met Councils' Regional Travel Demand Forecasting Model (predecessor to the new ABM). In addition, all responders had heard of or tried additional software packages such as Aimsun (TSS), Dynameq (INRO), DynusT, and TransModeler (Caliper), but were seen as too expensive in terms of training and licensing to effectively implement in most cases. Due to these drawbacks, the interviewees almost unanimously indicated that having the freedom to choose the best simulation software for each project was important since it would allow them to match the most appropriate software with their team's expertise. All of this was compounded by the fact that besides using CORSIM for many freeway modeling projects, no project forced the use of a particular software and between Voyager, Synchro, and Vissim most projects in the Twin Cities could achieve acceptable results.

Of the four core concepts discussed with the interviewees the idea of control (traffic signals) in the simulation network generated the most diverse and longest debate. The most defining problem with implementing control identified, was the lack of a unified database of current or even old timings. In addition, the sheer number of different controllers currently deployed and different parties that own them make it infeasible for any firm to collect unified signal timings. It was also revealed by one interviewee who had extensive experience in signal timings and the signal controllers currently deployed throughout the twin cities that just obtaining the data from the cabinet can be difficult in some cases. It was revealed that some cabinets do not even interpret inputs the same as others. An example of this occurs in the interpretation of yellow time, some controller's use a percentage of red time to define yellow time as included in the red time while others need it explicitly defined. These types of difficulties can vary and often the engineer needs to read each controllers individual manual and assemble them into a universal format and even though the difficulties exist for assembling signal timings it is often unavoidable. Therefore, all the responders would generally avoid collecting actual signal timings if possible due to the extensive time needed to convert them into a format that would import into a simulation packages. Many of the interviewees responded by saying that they would use Synchro to generate signals for their network based on tuning movements, using simple green splits, or simple pre-timed patterns if their focus was not on a particular intersection and instead on the network as a whole.

Due to the lack of regional sized model experience interviewees were questioned on how they would collect demand data for the Twin Cities region. For most responders the Regional Travel Demand Forecasting Model (RTDFM), maintained by the Metropolitan Council, would most likely be used to generate origin/destination (OD) matrices for the sub-areas defined by the user. In a few cases some responders had used other forms of OD estimates such as AirSage[3] to produce trip matrices for previous models but noted that for a network the size of the entire Twin Cities region would be cost prohibitive.

After affirming that the majority received OD trip matrices from the RTDFM another question was asked into whether those trips were modified in any way. Almost every respondent had their own "proprietary" way of adjusting the demand into smaller time pieces since the RTDFM generally has 1-hour demand slices or calibrating them to increase/decrease OD's as need. The 1 hour slices however are not refined enough for DTA models since they do not allow for peak spreading and generally the 1 hour original RTDFM demand periods would be reduced down to, 15/30-minute demand periods. It should be noted that with the introduction of the Met Councils new ABM model the time resolution of the demand is now in 30 min intervals for passenger vehicles. To calibrate the OD demand, it was compared and adjusted against freeway detector counts taken from MnDOT through the use of automated "proprietary" software to better match. Interviewees noted that this method of OD

[3] http://www.airsage.com/Industries/Transportation/Trip-Matrix/

adjustment would become difficult as more arterial streets were added due to the fact that there is little to no "free" data on those sections to compare model results against.

The final core topic, calibration, was closely tied to what the models purpose was and what sort of real data could be collected within the scope of the project. If the project allowed, real data that would be used for calibration included queue lengths, turning counts, saturation flows, vehicle distribution in lanes (staking evenly at signals), processing volumes, and freeway data (counts, speed, flow, congestion location). An essential form of calibration data that all interviewees agreed was important was the freeway data provided by MnDOT's detector network. However, it is well known that the data must be "cleaned" before being applied to the simulation model to minimize outlier data. Generally, the freeway data was averaged over several weeks during the fall (October/November) in an attempt to be outside the construction season. In addition to averaging over multiple days nearly every local stakeholder identified broken or offline detectors and omitted them from the real data set if their location was unimportant to the projects purpose. In cases where a detector set was important and was broken during the initial time gathered, a few stakeholders identified that real data was generated based on surrounding detectors and historical trends for that location to construct a complete data set.

Since MnDOT's freeway data is free and readily accessible it is almost always used if the model includes those locations. Otherwise as was mentioned previously additional data may be retrieved depending on the projects focus area and needs. Most of the other data sources mentioned would require extensive man-hours in order to collect. Even getting previously completed turning counts from many of the local cities or municipalities can be extremely time consuming to collect and convert (often from PDF files) into useable input. Multiple stakeholders mentioned that Minneapolis has a system in place (Transportation Data Management System[4]) to consolidate this data but it is only good within the city's boundaries.

[4] http://www.minneapolismn.gov/publicworks/public-works_traffic-counts

CHAPTER 6: COMMERCIALLY AVAILABLE PRODUCT EVALUATION

Originally five commercially available software packages were selected for comparison. However, during this project it was found that many of the local stakeholders (when performing sub area analysis) use Vissim as their simulation software. The research team therefore sought to include it so features found within Vissim were compared against the previously selected programs even though Vissim is a microscopic simulator by design.

Another program that was not selected but is mentioned often throughout this report and used by nearly every stakeholder is Synchro. Synchro is a “Traffic Signal Analysis and Optimization” software developed by Trafficware. It is the most commonly used signal optimization and “signal database” software package used in the Twin Cities.

6.1 TRANSMODELER

TransModeler is an all in one traffic simulation suite developed by Caliper Corporation based in Newton, Massachusetts. TransModeler is able to perform a wide variety of traffic simulation methods including microscopic, mesoscopic, and macroscopic. It could be argued that TransModeler is a hybrid simulator by design since the fidelity of the model is chosen at the link/node level. All three fidelities can be simulated simultaneously following a few constraints.

TransModeler is currently on Version 4.0 and a fully functional version was obtained for evaluation. In addition, Caliper provided a first draft of the metro geometry to assist the research team in testing the speed and capabilities of the program. TransModeler is an all-inclusive license so all features within it are under one license and no additional licenses or modules need to be purchased.

6.2 AIMSUN

Aimsun is an integrated transportation modelling software suite, developed and marketed by Transport Simulation Systems (TSS) based in Barcelona, Spain. The most recent versions of Aimsun (V8+) include the ability to perform four-step travel demand modeling and simulation at the travel demand model, mesoscopic, microscopic and hybrid level. The hybrid level is a combination of mesoscopic simulation and user selected microscopic sub areas. A prior project has developed a hybrid model of the entire Twin Cities metro region and it has been thoroughly tested and uncovered most details and performance capabilities of the software.

Aimsun is currently on version 8.1 (June 2015) and comprised of 7 different license levels. Additionally, add-ons that can be purchased with some licenses to unlock additional features such as extra processing threads. For the purpose of this project the MTO academic licenses of Aimsun 8.1 Expert edition are used for evaluation.

6.3 DYNUST/DYNUSTUDIO

DynusT (Dynamic urban system for Transportation) is an open-source Mesoscopic Dynamic Traffic Assignment (Meso-DTA) program. DynuStudio is a commercially available graphical user interface (GUI) and data management system for DynusT developed by RST International Inc. based in Bellevue Washington. A model based on DynusT has been provided to use for testing but the research team hasn't succeeded yet in opening it. Working with tech support to get the issues resolved.

The latest version of DynuStudio (V1.0.7) was not available in a demo license however version 1.0.6 is and will be used for evaluation. DynuStudio has two levels: standard and advanced. The advanced includes transit, subarea cutting and scripting tools.

6.4 DYNAMEQ

Dynameq is a "mesoscopic" dynamic traffic assignment simulator developed by INRO based in Montreal, Canada. Dynameq was originally by all accounts a mesoscopic model but more recently they have begun to add microscopic approaches into the program. According to INRO:

> "*Dynameq's traffic simulation is often referred to as mesoscopic due to the larger scale of network that can be calibrated; however, it bears far more similarity to the detail and fidelity of a microsimulation than other mesoscopic approaches. Dynameq moves individual vehicles on lanes, with car-following models, gap-acceptance models and explicit signal timings.*"

Currently Dynameq is on version 3.1.0 (March 2015) and a demo version was obtained for evaluation. The model of a small town in Canada was also provided to assist in the testing and evaluation of the software. These licenses are sold on a tier based system where the size of the network (zones, nodes, and links) that is to be modeled determines the license price. There also is an annual fee based on a percentage of the licenses that entitles the user to unlimited support and new versions.

6.5 CUBE AVENUE

Cube Avenue is a Mesoscopic Dynamic Traffic Assignment extension of Cube Voyager developed by Citilabs based in Lafayette, California. Cube Avenue is designed to replace highway assignment in travel demand models where dynamic traffic assignment is needed. Since Cube Avenue is an extension of Cube Voyager throughout this report they are often referred to together since many of the base components behind Avenue are contained in Voyager. Both of these are also contained inside of Cube base/desktop which acts as the front end graphical user interface (GUI) and geographic information system (GIS) for all of Cubes modules. A Cube Avenue model of the Twin Cities metro is available but hasn't yet been used for testing due to time and effort limitations.

Cube Base (and Voyager/Avenue) are on version 6.1.1 (Aug 2014) and a fully working academic license is already maintained by the MTO and is used for the evaluation. Cube's licensing structure consist of modules that are bought that have an optional annual maintenance fee that entitles the user to upgrades and tech support.

6.6 VISSIM

Vissim is a microscopic simulator developed by the PTV Group based in Karlsruhe, Germany. The main reason for including Vissim in this report was due to input and knowledge obtained during task 2 from the local stakeholders. Vissim was mainly used to identify features that stakeholders found useful and evaluating them in the other software.

Vissim is currently on version 7.00-10 (June 2015) and a fully working academic license is maintained by the MTO. Each of PTV's products consist of a base product and additional modules that unlock nonstandard features such as integration with other software.

6.7 COMPATIBILITY MATRIX

The compatibility matrix featured in this section is based on a list of capability's and features identified by local stakeholders and the research team. In order to avoid unnecessary repetition each, one of these capabilities has a brief description indicating what would constitute a "Yes" answer in the table to follow.

1. Transit Lines – The ability to simulate/replicate transit lines (Bus, LRT, etc.) and their impact on vehicle traffic.
2. Turning Movement Demand – The ability to use turning movements as a demand base similar to Synchro.
3. Origin Destination Demand – The ability to use Origin Destination (OD) Demand whether it be by trips or flows.
4. Number of Traffic Generation schemes – This is the number of ways vehicles can be generated into the network during the course of the simulation (assuming OD based demand)
5. Origin Destination Matrix Estimation (ODME) – The ability to perform some Origin Destination Matrix Estimation using a built in tool.
6. Signal-controlled intersections- The ability to simulate signalized intersections (fixed, actuated, etc.)
7. Right Turn on Red (RTOR) – The ability to simulate individual turnings located at intersections as right turn on red.
8. Traffic Signal Synchronization Offset – The ability to input an offset for coordinated signal systems.
9. Traffic Signal Optimization – Having a built in tool that is able to perform signal optimization at a single intersection.
10. Coordination signal optimization - Having a built in tool that is able to perform signal optimization on a corridor or selection of signals simultaneously.

11. Transit Signal Priority (TSP) – Being able to implement some form of TSP using built in tools.
12. Traffic Signal Preemption – The ability to have signals that can simulate vehicle preemption (train crossing, firetrucks, police, etc.)
13. External Traffic Signal Controllers – Being able to allow external control logic (either software based or physical) to operate traffic signals within the network
14. Highway Capacity Manual (HCM) Intersection Level of Service (LOS) – The ability to perform a HCM based LOS analysis on an intersection using built in tools.
15. Signal Warrants – Being able to perform some, if not all, of the Manual of Uniform Traffic Control Devices (MUTCD) signal warrants at an intersection.
16. Individual Lanes – The ability to have individual lanes represented in the simulation.
    a. Managed Lanes – The ability to open and close a lane base based on a strict time of day schedule.
        i. Restrict Lanes by Class (HOV, Transit, etc.) – The ability to select a single lane and designate it as reserved for a specific vehicle class.
        ii. Reversible Lanes – The ability to natively simulate reversible lanes using built in tools.
        iii. Shared Center Left Turn Lane – The ability to have a shared center left turn lane between at least two lanes of opposing traffic.
    b. Dynamic Lanes – The ability to effect a lanes physical properties (speed, flow, capacity, etc.) through various times of the day or based on roadway conditions.
        i. High Occupancy/Toll (HOT) Lanes – The ability to simulate lanes that have a cost associated with them and actively route vehicles into them and have a scheme to deter or attract vehicles built in.
        ii. Incident Simulation – The ability to simulate an incident that causes a change the vehicle behavior in a link, segment, or lane.
        iii. Work Zones – The ability to simulate work zones and their impact on the road by either link, segment, or lane
17. Enroute – The ability for a vehicle to find a new shortest path to its destination (assuming OD demand) after it has left its origin.
18. Overtaking vehicles on rural road – The ability for a vehicle to pass another vehicle on a two lane road if the opposite lane has an acceptable passing gap.
19. Yellow Boxes (blocked Intersections) – The ability to prevent or allow vehicles to block an intersection
20. Restricted movements – The ability to restrict specific movements at intersections based on time. (e.g. No Left Turns between 4pm – 8pm)
21. Roundabouts/Rotaries – Having a separate logic for the behavior of a vehicle inside a roundabout.
22. Synchro 7 import – The ability to interface with Synchro 7 in some way either by import/export or direct link.

23. Different Classes (SOV, HOV, Bus, etc.) – The number of different vehicle classes that can be specified within the model.
24. Parallel Working environment – Any tools that allow more than one modeler to work on the same simulation project at a given time.
25. Network validation checks – Any tool that actively runs or can be manually initiated that checks some part of the network (Geometry, Demand, etc.) for errors in coding or location of potential modeling issues (Short sections, low speed, etc.)
26. Exporting Options – Any tools that export part of the model
27. Scatter Plot tools – The ability to use built in tools to produce scatter plots of data from the simulation.
28. Subarea Extraction tool – A tool that can be used to extract a portion of a network to create a smaller network from a larger one.
29. Subarea Traversal Matrix Generation – The ability to generate Origin Destination matrices for a subarea.
30. Multithreaded – The ability to use multiple threads or computers to reduce the computational time of the simulation model
31. Detectors – The ability to replicate lane level loop detectors.
32. Scripting (for automation) – Having a programing language built into the software that allows the modification/ automation of certain tasks (e.g. increasing the speed on all highway segments by 10%)
33. Random Seed – The ability for the user to specify the random seed being used by the random number generator in the program.
34. Link specific vehicle factors – The ability to calibrate the model at the link level by adjusting factors that impact a vehicles behavior in a specific link such as reaction time. This is in contrast to networks where all roads of a specific type are given the same parameters.
35. Traffic Flow – If a traffic flow model governs how vehicles travel through the network (Car-following, lane changing, etc.).
36. Route Choice Method – The ability of the simulator to choose a shortest path based on network condition and how many were available.
37. Pedestrian Crossing (Volumes) – The ability to model pedestrian crossing volumes at intersections to simulate delay induced by pedestrians crossing.
38. Ramp Metering – The ability to simulate/replicate ramp meters.

Software	TransModeler	Aimsun	DynusT/ DynuStudio	Dynameq	Cube Avenue	Vissim
Version Number	**4.0**	**8.1**	**1.0.6**	**3.0**	**6.1.1**	**7.0**
Transit Lines	Yes	Yes	Yes	Yes	Yes	Yes[1]
Turning Movement Demand	Yes[2]	No[3]	No	No	No	No
Origin Destination Demand	Yes	Yes	Yes	Yes	Yes	Yes
Number of Traffic Generation schemes	9[4]	6[5]	1[6]	3[7]	1[8]	NA[9]
Origin Destination Matrix Estimation (ODME)	Yes[10]	No	No	No	No	No
Signal-controlled intersections	Yes[11]	Yes[12]	Yes[13]	Yes[14]	Yes[15]	Yes[16]
Right Turn on Red (RTOR)	Yes	Yes	No	Yes[17]	No	Yes
Traffic Signal Synchronization Offset	Yes	Yes	No[18]	Yes	No	Yes
Traffic Signal Optimization	Yes[19]	No[20]	No	No	No[21]	Yes[22]
Coordination signal optimization	Yes[23]	No	No	No	No	No[24]
Transit Signal Priority (TSP)	Yes	Yes	No	No	No	Yes[25]
Traffic Signal Preemption	Yes	Yes	No	No	No	Yes[26]
Traffic Signal Templets	Yes[27]	No[28]	Yes[29]	No	Yes[30]	Yes[31]
Synchro 7 import	Yes[32]	Yes[33]	No[34]	Yes	Yes[35]	Yes[36]
External Traffic Signal Controllers	Yes[37]	Yes	No	No	No	Yes[38]
Highway Capacity Manual (HCM) Intersection LOS	Yes[39]	No[40]	No	No	Yes	No[41]
Signal Warrants	Yes[42]	No	No	No	No	No[43]
Individual Lanes	Yes	Yes	No[44]	Yes	No[45]	Yes
Managed Lanes	Yes	Yes	No	Yes	No[46]	Yes
Restrict Lanes by Class (HOV, Transit, etc.)	Yes[47]	Yes[48]	No	Yes	No	Yes
Reversible Lanes	Yes[49]	No	No	No	No	No
Shared Center Left Turn Lane	Yes[50]	No	No	No	No	No
Dynamic Lanes	Yes[51]	Yes	No	No[52]	No	No[53]
High Occupancy/Toll (HOT) Lanes	Yes[54]	No[55]	Yes[56]	No	Yes[57]	No
Incident Simulation	Yes[58]	Yes	Yes[59]	Yes[60]	No	No
Work Zones	Yes[61]	Yes[62]	Yes[63]	Yes[64]	No	No
Enroute	Yes	Yes	Yes[65]	No	No	Yes[66]

Software	TransModeler	Aimsun	DynusT/ DynuStudio	Dynameq	Cube Avenue	Vissim
Version Number	**4.0**	**8.1**	**1.0.6**	**3.0**	**6.1.1**	**7.0**
Yellow Boxes (blocked Intersections)	Yes[67]	Yes[68]	No	No	No	Yes[69]
Restricted movements	Yes[70]	Yes	No	Yes	No	Yes[71]
Roundabouts/Rotaries	Yes[72]	No[73]	No	Yes[74]	Yes[75]	Yes
Overtaking vehicles on rural road	Yes	Yes	No	No	No	No[76]
Different Classes (SOV, HOV, Bus, etc.)	Inf[77]	Inf[78]	3[79]	16	255[80]	Inf[81]
Parallel Working environment	Yes[82]	Yes[83]	No	Yes[84]	Yes[85]	Yes[86]
Network validation checks	Yes[87]	Yes[88]	No	Yes[89]	Yes[90]	No[91]
Exporting Options	Yes[92]	Yes[93]	Yes[94]	Yes[95]	Yes[96]	Yes[97]
Scatter Plot tools	Yes	Yes	Yes	Yes	Yes[98]	No[99]
Subarea Extraction tool	Yes	Yes	Yes	Yes	Yes	Yes[100]
Subarea Traversal Matrix Generation	Yes[101]	Yes	Yes	Yes	Yes	No
Multithreaded	Yes[102]	Yes[103]	Yes[104]	Yes[105]	Yes[106]	Yes[107]
Detectors	Yes	Yes	No[108]	No[109]	No	Yes
Scripting (for automation)	Yes[110]	Yes[111]	Yes[112]	Yes[113]	Yes[114]	Yes[115]
Random Seed	Yes	Yes	Yes	Yes	Yes	Yes
Link specific vehicle factors	No[116]	Yes[117]	No	Yes[118]	No	No[119]
Traffic Flow	Yes[120]	Yes[121]	Yes[122]	Yes[123]	No[124]	Yes[125]
Route Choice Method	Yes[126]	Yes[127]	Yes[128]	Yes[129]	Yes[130]	Yes[131]
Pedestrian Crossing (Volumes)	Yes[132]	No[133]	No	No	Yes[134]	Yes[135]
Ramp Metering	Yes[136]	Yes[137]	Yes[138]	No	No	Yes[139]

[1] Public transit is modelled as trams and/or buses in high detail with fixed routes. Stops and timetables are modelled as well. Waiting times at stops depend on the timetable and on a user defined random distribution for the passenger interchange times. Optionally the dwell time calculation method can be chosen which uses real passenger flows to determine waiting times.

[2] In TransModeler turning movement-based demand constructs a vehicle trip by solely relying on turning volumes. The base of turning movement-based demand in TransModeler is a dynamic turning movement table that specifies the flow for each turning movement in each time interval. TransModeler will simulate either static or dynamic turning movement counts and can simulate a combination of one or more turning movement tables and trip matrices.

[3] Aimsun does have a secondary form of demand called a "Traffic State" that allows the user to specify a flow on every input section with turning percentages at each intersection. One could potentially convert turning movement flows into these percentages so a turning movement based demand could be run by Aimsun.

[4] TransModeler has three time distribution rates (Constant over Time, Curve-based, Time-Dependent Matrices) and three Headway Distribution Rates (Deterministic, Random [uniform], Random [Negative Exponential]) which when combined give a total of 9 different combinations for travel generation schemes.

[5] Aimsun refers to the generation rate of vehicles as "Traffic Arrivals." In total Aimsun has 6 different arrival schemes: Exponential, Uniform, Normal, Constant, External, and ASAP. The External arrival scheme requires the use of the Aimsun API and a defined external Dynamic Link Libraries (DLL) in order to function. It is also noted that there is one global scheme for each experiment and when OD demand is used specific OD pairs may be assigned a different arrival scheme.

[6] When using OD based demand matrices DynusT will schedule a departure time for each vehicle during the time-of-day profile specified in a "random distribution manner."

[7] Three methods are available:

- Poisson-the Poisson traffic generator produces vehicle departure times as a Poisson Process, in which case both the number of vehicles generated and the departure times are random. The number of vehicles generated follows the Poisson distribution, while inter-departure times (the duration of time between two sequential departures) follows the negative exponential distribution. Because the variance of the Poisson distribution is equal to the mean, the variability in the number of vehicles generated using this method may in some cases be higher than desired.

- Conditional-the Conditional traffic generator has almost no variability in the number of vehicles generated. For each O-D pair, for each table of the O-D matrix, the number of vehicles generated is obtained by multiplying the flow rate by the duration of the corresponding matrix interval. The only variability in the actual number of vehicles generated is due to the procedure by which the real-valued result of this product is rounded to obtain an integer number of vehicles. A bucket-rounding procedure is used which ensures that the total number of vehicles generated at an origin for a specific matrix interval is equal to the sum of the flow rates (row sum) multiplied by the duration of the interval (rounded to the nearest integer). The inter-departure times follow the negative exponential distribution, as when using the Poisson traffic generator.

- Constant-the Constant traffic generator produces the same number of vehicles as the Conditional traffic generator. The inter-departure times are constant, and equal to the duration of the matrix interval divided by the number of vehicles generated. Only the first departure time for each interval and each O-D pair is random, following a uniform distribution over the duration of the inter-departure time.

[8] In Cube Avenue the departure time of a packet (a collection of vehicles) in a given time interval is randomly selected based on the random seed.

[9] When using Origin-Destination matrices as an input for the Vissim network there does not appear to be anyway to control the arrival rate of vehicles as seen in other simulation packages. The only mention of vehicle arrival rates in the Vissim user manual is under the explanation of the “Random Seed” where it say “This, e.g., allows you to simulate stochastic variations of vehicle arrivals in the network.” Based on this statement it would seem to infer that the vehicle generation rate is random.

[10] The O-D Matrix Estimation (ODME) procedure in TransModeler is based on the work of Nielsen (1993, 1998), who independently developed it as a procedure for TransCAD 2.1. The method was re-implemented by the Caliper Corporation. The ODME procedure is meant to help adjust OD demand to match available counts and overall improve the simulation results by reducing over or underestimated demand from wherever the Origin Destination matrixes originate from (Typically a travel demand forecasting model). This method has the advantages of treating counts as

stochastic variables, as well as working with any traffic assignment method. It therefore is attractive for use with the Stochastic Assignment method, as well as with a Dynamic User Equilibrium Assignment. There are three types of matrix estimation methods in TransModeler: Single Path, Multiple Path and Gradient. Single Path and Multiple Path O-D estimation are based on volume-to-count ratios calculated along shortest paths. In the Single Path option, the assigned volumes along a single best path for each O-D pair are compared with the counts, and the O-D flows updated accordingly. In the Multiple Path option, the assigned volumes along multiple shortest paths are compared with the counts, and the O-D flows updated. The Multiple Path method is an enhancement of the Single Path method and can yield more accurate results. The Gradient method assesses the contribution of each O-D flow to an overall objective function that must be minimized. Assignment and O-D trip updates are repeated in an iterative loop until a convergence criterion is reached or for a user-specified number of iterations, whichever comes first. The Gradient method is more appropriate when there is high confidence in the prior estimate for the O-D matrix.

[11] When a signal is modeled in microscopic simulation, signals are explicitly modeled based on their parameters. In mesoscopic regions, intersection lane level capacities are determined based on HCM 2010 methods.

[12] When a signal is modeled in microscopic simulation, signals are explicitly modeled based on their parameters. When modeling in the mesoscopic simulator, traffic signals are simulated by using events that allows or forbids vehicles to enter into the junction. When the traffic light is changed to red all vehicles waiting to enter into its downstream junction are not allowed to proceed until the next “green” event. When the traffic light is changed to green then the node server looks for the next vehicle that can enter into the junction considering the new permitted movements.

[13] DynusT has the ability to perform Pre-timed or actuated signals. Pre-timed signals use a combination of fixed user-defined parameters including: cycle length, phasing plans, green time, and amber time. Actuated signals use DynusT to determine the best green times for a phase according to the approach volumes and the user only has to specify the max and minimum green times.

[14] Only one-ring controllers can be modeled explicitly, all others would need to be converted into equivalent one-ring timings.

[15] Four signalized intersection types are offered in Cube Voyager/Avenue: signal saturation flows, signal geometric (HCM), adaptive signal saturation flows, and adaptive signal geometric (HCM). They are defined as follows taken from the Cube Voyager Reference Guide.

- Signal, Saturation Flows model: It is developed to model capacities, queues, delays and LOS at fixed time signal controlled isolated intersections. The user inputs include geometric characteristics of the intersection, signal timing arrangements, and demand flow information. The methodology is based on the Catling's delay method and the TRRL (Transport and Road Research Laboratory, UK) Report 105.

- Signal, Geometric (HCM) model: It addresses the capacities, queues, delays and LOS for lane groups and the LOS for intersection approaches and the intersection as a whole at signalized intersections. Each lane group is analyzed separately in the HCM model. It considers a wide variety of prevailing conditions, including the amount and distribution of traffic movements, traffic composition, geometric characteristics, and details of intersection signalization. The methodology is based on the HCM 2000 signal model.
- Adaptive Signal, Saturation Flows model: It is an advanced model based on the Signal, Saturation Flows model. The model optimizes the signal timing based on the intersection geometric characteristics, signal parameter bounders and demand flow information. The methodology tries to iteratively fit delay parabolas based on three distinct and reasonable signal timing plans, i.e. phase timings and cycle time, and picks the plan at the minimum delay point, until no delay reduction could be reached. The delay is calculated by the Catling's delay method. The methodology was inspired by the TRANSYT model.
- Adaptive Signal, Geometric (HCM) model: The model is similar as the Adaptive Signal, Saturation Flows model, except the delay is calculated by the HCM 2000 method.

[16] Vissim has many external plugins that allow the control of signalized intersections using a variety available plugins. These plugins include signal controllers such as Economize ASC/3, Ring Barrier Controller, and VAP (vehicle actuated programming) controllers. It also has the ability to use an external signal control interface such as SCOOT, SCATS, or LISA + OMTC.

[17] In Dynameq you can only select RTOR by single plan, either all right turns are RTOR or none are.

[18] DynusT can only apply offsets for pre-timed signals. Actuated signals use DynusT to determine the best green times for a phase according to the approach volumes and the user only specifies the max and minimum green times.

[19] The signal optimization methods include the Webster’s Method and a Stochastic Simultaneous Search. Webster’s Method is a simple algorithm that seeks to minimize lost time at the intersection. The Stochastic Simultaneous Search algorithm allows you to minimize one of a set of measures of effectiveness (MOE) (delay, delay and stops, fuel consumption, queue length, or a weighted combination of the MOEs).

[20] Aimsun does not have a tool for performing system wide optimizations of traffic signals. Aimsun does have a tool called the “Control Plan Generator” that can generate signal timing plans for a node or a selection of nodes based on the turn capacity and turn assigned volumes and some user specified parameters. (This tool requires an Aimsun Professional for Travel Demand Modelling, Aimsun Advanced or Aimsun Expert Edition license.) Aimsun’s recommended signal optimization solution is to use the Synchro import/export tool to transfer a network between the two programs and perform the signal optimization in Synchro.

[21] Voyager does not have a signal optimization routine built in however it does have a utility that imports Synchro data and converts it to Cube Voyager/Avenue format.

[22] Vissim can optimized fixed time signal controllers using the VISSIG add-on module or can also import/export data to and from Synchro 7 (which can optimize the signal timings) using an add-on module. In addition, PTV has another product called Vistro that can import a Vissim network and optimize signal timings.

[23] After the intersections in the coordinated system have been identified and the coordination parameters at each intersection defined, the coordinated signal timings can be optimized.

Phase splits are optimized at each intersection for every evaluated cycle length using the process described in *Signal Optimization.* The optimal cycle length and offsets are decided by minimizing the Performance Index (PI), which is computed from a weighted sum of the following four measures of effectiveness (MOEs):

- Average control delay, as defined by the Highway Capacity Manual
- Average queue length by lane
- Average number of stops
- Arrival on Green (AOG%) percent

[24] Vissim cannot perform coordinated signal optimizations internally. It can however use an add-on module to import/export to Synchro 7 or can use PTV’s own Vistro (separate license) to optimize signals.

[25] Priority on signal control junctions can be modelled completely with detection and control logic using the Ring Barrier Controller add on.

[26] Vissim can perform signal preemption using the Ring Barrier Controller add on.

[27] TransModeler allows an unlimited amount of signal templates to be defined to aid in the programming of multiple signals that follow a predictable pattern.

[28] Aimsun does not have a built in templets for traffic signals however scripts can be written in the scripting language of Aimsun (python) to automate traffic signal importation or to generate templets.

[29] Since DynusT only has three different styles of traffic signals with a few user defined variables each they are easy to replicate and apply to other nodes.

[30] Given that Cube Voyager/Avenue is script (text) based generic signal plans can be saved and edited to fit individual intersections.

[31] While Vissim does not have any signal templets it does have the ability to import timings in some add-on modules. Specifically, the RBC (Ring Barrier Control) add-on allows the user to import the signal timing from a file. In this way a series of "templets" could be created for a region where signal timings are similar or follow a pattern.

[32] TransModeler can import Synchro 6 as well but this feature will be removed in future versions.

[33] Aimsun can only read and export the Universal Traffic Data Format (UTDF) defined by Trafficware (Synchro V.7+).

[34] The DynuStudio Manual does mention that signal timings can be imported from an HCM based software such as Synchro but no tool is readily available in DynuStudio/DynusT.

[35] Cube can import Synchro 7 data using a built in import utility.

[36] Vissim has an add-on module that imports/exports synchro 7 networks.

[37] External traffic signal controllers can be implemented through the use of the GISDK scripting language built into TransModeler.

[38] External controllers are available through a series of add-on modules. The available modules are: LISA + OMTC, SCATS interface, SCOOT interface, or an External Signal Control not mentioned but the executable (*.exe) and the program libraries (*.dll) are available and developed under the PTV application program interface (API).

[39] TransModeler can calculate the Level of Service based of the 2010 Highway Capacity Manual. However, a turning movement table with hourly volumes is required.

[40] Aimsun does not have a built in tool for analyzing the Level of Service (LOS) for a particular intersection. Instead Aimsun suggests to use the SYNCHRO import/export tool to run a LOS analysis in SYNCHRO of an Aimsun network. Version 8.1 did introduce some HCM 2010 calculations but these are limited to approaches, weaving areas, and merge/diverge areas.

[41] Vissim cannot perform HCM calculations internally. It can however use an add-on module to import/export to Synchro 7 or can use PTV's own Vistro (separate license) to perform HCM calculations.

[42] Given a Turning Movement table containing hourly volumes TransModeler can evaluate warrants 1-3 at an intersection based on the 2009 Manual of Uniform Traffic Control Devices.

[43] Vissim cannot perform signal warrant calculations internally. It can however use an add-on module to import/export to Synchro 7 or can use PTV's own Vistro (separate license) to perform warrant analysis.

[44] DynusT does not construct individual lanes as part of its model. However as is often done in forecasting models certain lanes, such as HOV lanes, can be simulated by using a separate parallel link. This link cannot transfer traffic between other links unless it is through a node and therefore can make it difficult to replicate open access systems where vehicles can enter at any point.

[45] Cube Voyager/Avenue does not construct individual lanes as part of its model. However as is often done in forecasting models certain lanes, such as HOV lanes, can be simulated by using a separate parallel link. This link cannot transfer traffic between other links unless it is through a node and therefore can make it difficult to replicate open access systems where vehicles can enter at any point.

[46] Cube Voyager/Avenue has the ability to change the attributes of links with in a time segment specified but is hardcoded and difficult to program multiple scenarios. For example, consider a model that is 240 minutes long and split into 30 minute segments. A links attributes can only be changed within a time segment and therefore would be limited to 30 min intervals. Furthermore, if either the time interval or duration of the model was altered the change in the links attributes would not start at the correct time. This is due to the fact that the segment from which the change occurs in is referenced by the number of segments from the beginning of the simulation.

[47] Lanes can be reserved for specific vehicle classes at all time or can be changed through the use of the "Traffic Management toolbox."

[48] Lanes can be reserved for specific vehicle classes at all time or can be changed through the use of the "Traffic Management" strategies and policies.

[49] One-Way directional links can be marked as reversible and the direction of flow can be managed using the "lane access control" in the Traffic Management Toolbox.

[50] Any bidirectional link with at least 3 lanes combined can be marked as having a shared left turn lane that can be used by either direction of the single bidirectional link selected.

[51] TransModeler has a few built in toll models including Fixed, Time-Repentant, and Traffic-Responsive. It also can use a User-Programed toll to accommodate any toll pricing logic

[52] HOT Lanes and Toll Lanes cannot be modeled explicitly but can be added into the generalized cost function.

[53] Vissim does not have an easily accessible GUI for programing dynamic lane facilities. While such attributes could be implemented through the "event" scripts these are not as intuitive and require more effort to set up.

[54] Dynamic Lanes can be replicated through the use of TransModeler's "Lane Use Sings"

[55] While lanes can be reserved to "HOT" vehicles there is no built in tool for easily accommodating a detailed HOT lane with a pricing scheme.

[56] Hot lanes can be simulated in DynusT but they are limited to closed access systems. Since DynusT does not represent individual lanes it therefore can only transfer HOT/HOV vehicles from general purpose lanes to the HOT lane at specific nodes. It should be noted that during the coding of the HOT lane the corresponding link running parallel is specified. Internally a "powerful algorithm" is used to divert only a portion of HOT vehicles while maintaining a user specified speed in the HOT lane.

[57] Hot lanes can be simulated in Cube Voyager/Avenue but they are limited to closed access systems. Since Cube Voyager/Avenue does not represent individual lanes it therefore can only transfer HOT/HOV vehicles from general purpose lanes to the HOT lane at specific nodes. The user would need to construct a cost function that will operate the HOT lanes.

[58] Incidents are limited to the speed impact on each lane. The user defines a time interval over which it will take place, the length of the incident in the segment (per lane), and the speed impact on each lane.

[59] DynusT allows the simulation of incidents at the link level by reducing the capacity of the link by a percentage for a specified time interval.

[60] Can be done through the use of “event times” but is limited to lane level and link level events. By lane events can only prohibit lane use by class and links (which effect all lanes within it) can only impact the free flow speed, effective length factor, and response time.

[61] Work zones are implemented the same as incident simulations and comprise of a user defined time interval over which it will take place, the length of the work zone in the segment (per lane), and the speed impact on each lane.

[62] While Aimsun does not have a “work zone” strategy the effects of a work zone can be implemented though a combination of all the other available strategies.

[63] DynusT can replicate work zones at the link level between specified time intervals through the use of three parameters: a capacity reduction rate, a new posted Speed, and the maximum flow rate through the work zone.

[64] Dynameq does not have a specific “work zone” menu but many of the effects can be implemented via the “event times” in the same way incidents can be modeled.

[65] En-Routing can be broken done by class as a percentage of vehicles that are allowed to enroute, furthermore an indifference band (representing the inertia for switching to a new route) and threshold bound (the difference between the current selected travel time and newly recommended route travel time) must be met before a vehicle will enroute during a simulation.

[66] Vissim refers to en-routing as “Route Guidance.” It consists of multiple factors that influence the vehicles effected and when they look for a new path. Vissim allows two difference “guidance” systems to be specified where the parameters “Route guidance interval” (the time between shortest path search from current location) and “Offset” (Duration of processing times and run times of messages in real route guidance systems). Each guidance system may be associated with as many vehicle types as needed. For example, if the vehicle type “car” is chosen to have a guidance system all vehicles of that type will.

[67] Known as Driver Compliance. When vehicles proceed into intersections when their downstream lanes are full, they may have to stop inside the intersection, blocking vehicles making conflicting movements. You can specify a compliance rate for both controlled intersections (i.e. intersections controlled by stop signs, yield signs or traffic signals) and uncontrolled intersections. Further, you can choose whether drivers will respect the intersection blocking rule in specific circumstances. By default, drivers making turns that do not cross other movements (e.g., a right turn from the right-most lane to the right-most downstream lane) will not comply with the rule, irrespective of the specified compliance rate. Additionally, you can indicate whether you want vehicles to comply with the rule, at the specified rate of compliance, only if there are other vehicles present at the intersection, vehicles that the subject vehicle might potentially block, when the subject vehicle arrives.

[68] Yellow Boxes can be set individually for each intersection. The yellow box speed controls when a vehicle will enter the junction node. In a microscopic simulation, a vehicle approaching a yellow box junction will avoid entering the junction area whenever the preceding vehicle is moving at a speed below this parameter (in km/h). This speed is defined for each turn and in effect, the yellow box behavior can be deactivated by setting it to 0 for a turn. In mesoscopic vehicles cannot block intersection but must wait for a minimum safe headway and be able to fully complete the turn within that time before leaving their previous section.

[69] Yellow boxes can be modeled using conflict areas and priority rules to either prevent vehicles from blocking an intersection or promoting them to.

[70] Lanes may be closed to specific vehicles but turning movements within an intersection cannot be restricted with built in tools (e.g. No Left turn between 4-6pm).

[71] When using Dynamic Assignment, connectors may be closed to certain vehicle classes at specific time intervals

[72] TransModeler includes a tool included to create geometrically correct circular roundabouts as well as different logic for controlling the behavior while inside the roundabout.

[73] An import tool exists to help form roundabouts but does not contain specific logic that is differentiates the roundabout from other roads. They are merely comprised of short sections linked by nodes.

[74] Roundabouts are explicitly labeled and a series of user selectable parameters govern them differently than regular links.

[75] Cube Voyager/Avenue has two specific ways to model roundabouts: a Gap-acceptance model (Each entry is characterized by a critical gap and a follow-up time, HCM 2000) and an Empirical model (Each entry is characterized by a capacity slope and a capacity intercept.)

[76]Vissim uses directional links, therefore it is only possible for vehicles to overtake at fixed locations using priority rules.

[77] The vehicle specifications in TransModeler are very diverse and highly detailed. TransModeler uses a “Vehicle Fleet” to specify what the percentage of vehicles (Trucks, Passenger cars, Motorcycles, etc.) make up the fleet. There does not appear to be a limit on the amount of “classes” one is able to specify. However, the sum of the percentages of each class should equal 100. In addition, each vehicle can also have a secondary label called a “Category.” These categories are used to determine lane use behaviors together there are 5 vehicle categories: Truck, Electric Toll Collection (ETC), Probe Vehicle, User A, User B. Another aspect that is handled separately is the occupancy of each vehicle. This will determine whether they are HOV vehicles or not. The occupancy of a vehicle can either be explicitly defined in an OD matrix or it can be randomly chosen based on a user defined mean and variance of each vehicle in the vehicle fleet.

[78] In Aimsun each vehicle is defined separately and can be included in more the one “vehicle class.” For example, an HOV vehicle can belong to both an HOV class and a HOT class so any HOV vehicle could use either facility limited to those classes. There is no apparent limit on the number of vehicle and vehicle classes that can be created.

[79] DynusT can only have three predefined classes: HOV, PC (Passenger car), and Trucks

[80] Cube generally refers to vehicles classes as modes.

[81] In Vissim they are referred to as vehicle categories and there does not appear to be a limit on the number that can be modeled.

[82] TransModeler has a built in tool to merge two simulations projects containing road networks, turning movements, and signals timings into one simulation project.

[83] Aimsun provides a few different tools that help with parallel working environments. The first is the “Revision” system in which a base network can be split into multiple revision networks. These networks can then be consolidated into full networks or consolidated back into the base. The catch is that when a revision is consolidated into a base it will affect all other revisions created from that base. The second tool is the ability to copy-paste most network properties between to open Aimsun networks. This like the revision system has its own limitations such as not being able to transfer signal timings between two networks when the intersection is copied into another network.

[84] Merging networks: A merge operation consists of replacing a subarea of the network with a network from an external source scenario.

A merge operation works as follows:

- The portion of the current network defined inside the selected subarea is removed, including the gate links that traverse the subarea polygon boundary.

- The entire network from the source scenario is put in its place. The external gate links of this scenario should have a 1-to-1 correspondence with the gate links of the subarea that was removed from the current scenario.

- The gate links of the source scenario are then attached to the corresponding nodes in the current from which the gate links of the subarea were removed.

[85] That base networks in the Cube architecture are based on a GIS framework so users can work in any GIS environment to construct the network. (ArcGIS is included with Cube licenses). Also since Cube Voyager/Avenue is script/text based multiple users can be working on different scripts and merely copy and paste them from one network to another.

[86] Vissim introduced the ability to copy/paste network elements from one network to another.

[87] TransModeler has an extensive network checker that can be run at any time and includes intersection geometry, segment length, lane connectivity merging conflicts, segment geometry, model fidelity, and missing signal timing plans.

[88] Aimsun provides a “Check and Fix” option in the context menu of any experiment. This tool will check a variety of items that the selected experiment may depend on. These include network errors like floating section, speeds equal to zero, no signal plan specified, etc.

[89] Dynameq has a number of network validation checks, which can be very helpful in identifying coding errors in the network. These are executed whenever a scenario is opened and whenever data is imported into a scenario (from Dynameq network files or shapefiles). A validation check can generate either an error message or a warning message. An error message indicates a problem that must be resolved before Dynameq will run a DTA using the network. A warning message indicates a potential problem, but it is up to the user to judge if the problem is real or not.

[90] A few programming issues can be identified by Cubes "_FLAG" system. This flag tool will check the network for items like: dangling links, small gaps between nodes, and unused nodes.

[91] While there is not a specific network checker tool, Vissim does have certain menus (such as the Ring Barrier Controller) that have active log windows reporting errors, warning, and messages. These logs are actively warning the user of incorrect or misprogrammed signal parameters.

[92] TransModeler makes it possible to export a map layer to many common interchange file formats: Standard Geographic File (.dbd), Compact Geographic File (.cdf), Text/Geography (.geo), Atlas BNA, AutoCAD DXF, Esri Shapefile (.shp), Esri Ungenerate (.lin, .pts), MapInfo Interchange (.mif), Oracle Spatial Layer, Google Earth Document (.KML), Google Earth Compressed Document (.KMZ), SQL Server Spatial Layer, Esri File Geodatabase (.gdb), Esri Feature Class, SDTS Point. It can also export Synchro Volumes if the networks are consistent between each other.

[93] Aimsun has the ability to export to 5 different files types: GIS (ESRI Shape file, MapInfo, GML), Google Earth, Image File (PNG), Network as 3D file (.3ds, .obj, .osg, .ive), and Synchro (UTDF). Some other data, such as OD matrices, can be exported individually in various formats.

[94] Various items in DynusT can be exported to CSV (Comma Separated Variable), DBF (Database File), Excel, EMME2, and Google Earth files.

[95] Dynameq supports exporting to: Dynameq network files, Shapefiles, An Emme network file, or Synchro Version 7 files.

[96] Cube can export as GIS (geographic information system) shapefiles, layers, or networks that can be read by most GIS based software.

[97] Vissim only includes exporting to Visum and 3ds Max by default. Other export options such as to Synchro 7 are available as add-on modules.

[98] In addition to scatter plot tools multiple tools are available to generate reports that auto populate themselves based on new results.

[99] PTV has traditionally used Excel as its plotting tool. However, PTV has been integrating plotting tools into Vissim and in version 7 they introduced bar and line charts within Vissim.

[100] The subarea extraction tool in Vissim is limited in that the user selects the network elements they wish to be in a sub network and "saves" them to a new file.

[101] It also outputs the average travel time and distance traveled through the subarea.

[102] TransModeler has the ability to use multiple threads in many of its operations. There does not appear to be a limit on the number of threads that can be used.

[103] Different editions have different thread limitations. The expert license (highest) contains 8 and additional threads can be purchased from TSS (Aimsun's parent company) with no limitations on the number of threads that may be purchased.

[104] When loaded DynuStudio automatically found and set the maximum number of Threads on the Computer as the default. There does not appear to be a limit on the number of threads that can be specified.

[105] Number of threads appears to only be limited by the computer the simulation is running on.

[106] With Cube Cluster (available as a separate license) enables multiple threads and processers even over several computers to be used to reduce model run time. It is also noted that Cube 6.4 (released June 2015) is now 64-bit so significant speed improvements have been seen over the 32-bit version.

[107] Vissim is multithreaded and is only limited by the number of computing threads available to the computer it is installed on. Vissim is also able to, in version 7, distribute multiple runs over a number of computers and produce results from each run individually or as a whole.

[108] Detectors are only used in ramp meters to facilitate a dynamically changing ramp meter. It should be noted that since DynusT does not have individual lanes, detectors are not very useful since they would only record statistics on the link they span which are aggregated already.

[109] "Lane Detector" output is an option for a given simulation. However, these outputs are taken at the downstream end of the link.

[110] TransModeler has a built in application programming interface (API) which they refer to as the GISDK. The GISDK

[111] Aimsun employs both a built in python based scripting language that can perform most scripting based needs but also has a platformSDK that can be used to do virtually anything an experienced user/programmer can conceive. The platformSDK does require a separate license in order to run whereas the base python scripting libraries is included with all licenses.

[112] DynuStudio uses a scripting language called "UScript" which is based on the Python. This scripting allows user to access and manipulate DynusT formatted files.

[113] The Dynameq API allows users to automate Dynameq procedures and to read and write Dynameq data using the Python programming language. The Dynameq API allows specification and automation of DTA procedures and in-memory access to time-varying network and demand (matrix) data, traffic control plans, simulation results, and other project data.

[114] Since Cube Voyager/Avenue is based of scripts it has a built in language to perform all basic and complex functions.

[115] Scripting can be done through the add-on module programming interface (API) which is not included with all Vissim licenses. The scripts themselves can be in one of three languages: Visual Basic (*.vbs), JavaScript (*.js), and Python (*.py).

[116] Some level of localized link specific parameters can be specified through the use of the "Traffic Management Toolbox" to influence the speed of vehicles though a section. Free Flow Speed, Capacity, Lane Capacity, and Lane Saturation Flow rate can be assigned individually if a valid field is added to each link and selected in the "Projects Settings" menu.

[117] Aimsun has multiple parameters that can be modified at the link/lane level that impact the macro, meso, micro, or hybrid simulations. For a mesoscopic model these included: Speed Limit, Jam Density, Reaction Time Factor, User defined costs, and two additional option for the lane selection model in penalizing shared and/or slow lanes.

[118] Effective length and response time can be modified by link for calibration.

[119] The only link level variables available for calibration is the cost per km, surcharge 1 and surcharge 2 that is used during path search. These variables are part of the general cost equation. The first two parameters, cost and surcharge 1, are impacted by a vehicle specific gamma variable. While surcharge 2 is simply added to the general cost of any vehicle that takes that link.

[120] In TransModeler's microsimulation model there are a total of 4 modeling "groups" that control driver behavior: Acceleration, Lane Changing, Merging and Yielding, and Response to Traffic Controls. The mesoscopic model uses a speed-density relationship to model vehicles in those regions and aggregates them into cells (car-following) while on long stretches and streams when they approach tunings. While lane changing is not specifically modeled vehicles can move to a different lane based on a gap model and overtake other vehicles.

[121] Aimsun employs different versions of Traffic Flow models based on the fidelity chosen. In general the components that make up the mesoscopic simulator consist of two models. The first is a model for vehicles in sections in which two sub models, car-following and lane-changing, are applied. The second is for when vehicles are within a node that consist of three different sub models. These sub models are: a gap-

acceptance model, a lane change model, and a turning model. Individual vehicles and lanes are maintained in the mesoscopic and hybrid simulator.

[122] DynusT is based on the Anisotropic Mesoscopic Simulation (AMS) model which is a modified Greenshield model. Two default models are available, the first is for links with uninterrupted flow and high capacity while the second is for links with interrupted flow and lower capacity. There is no limit on the number of traffic flow models that can be specified. However there can only be one traffic flow model per link class (freeway, arterial, etc.)

[123] The three components of Dynameq's Traffic Flow model are car following, gap acceptance, and lane changing.

[124] Cube Avenue is an extension to the Voyager highway assignment and therefore does not include anything that could be referred to as a traffic flow model since there are no individual vehicles to control. Packets (collections of vehicles headed to the same destination) are created at each origin for each of the pre-specified time segments from their start times and start moving along their routes. As they leave each link their flows are added into the network's volume fields behind them. They move according to the travel time in the link and turn as well as any additional delay occurred in the junction. Upon trying to enter a link they first check to see if they can fit according to the links storage and current amount of vehicles, if they cannot fit they are blocked and wait in their current link. The travel time is a user defined formula to compute the congested travel time after each interval (the first interval uses free flow travel time).

[125] The traffic flow model in Vissim is based on three separate models. The first model is the car following model that was developed by Prof. Rainer Wiedemann in 1974 and 1999, either one of these models may be chosen per road type. The second is a lane change model which is highly customizable per road type and finally a lateral model that simulates the cars movement within its current lane.

[126] TransModeler has three available methods for route choice: Deterministic Shortest Path, Stochastic Shortest Path, and Probabilistic Route Choice. The simplest method is the deterministic shortest path, whereby all vehicles follow the absolute shortest path. The stochastic shortest path method is similar to the deterministic shortest path in that all vehicles choose a shortest path. Path costs are randomized, however, for each individual vehicle to account for variations in perception and behavior. Thus, there is not one, but many, shortest paths between a given origin-destination pair. Finally, the probabilistic route choice model uses a multinomial logit model (MNL) choice model to simulate a driver's choice among a set of alternative paths, each having a utility that describes its relative attractiveness. In addition to the route choice models above there are several "options" that apply to all of the models and can be set by the user. These options effect the route choice models by

introducing parameters such as: Turning Delays, En-routing after unexpected delay, generalized costs, Freeway Transfer Penalties, HOV travel time factors, and HOT lane travel time factors.

[127] Aimsun's Route Choice is split between the shortest path algorithm and path selection. Aimsun has several user defined parameters to specify how the shortest path algorithm will work. These include how many shortest paths will be selected at the beginning of the simulation, the maximum number to keep in memory, and the maximum number of paths to consider per interval. Once the shortest paths have been calculated path selection is done through one of the following route choice models: fixed based on free-flow travel times, fixed based on travel times after warm-up period, binomial, proportional, logit, C-logit, or user-defined.

[128] At each assignment iteration of DynusT one of the two assignment algorithms is run. The user has the choice of using either the traditional MSA algorithm m or the new GFV (Gap-Function Vehicle). The GVF algorithm is selected as the default and has two additional user definable parameters: the GFV scaling factor (impacts the number of vehicles that switch paths at each iteration) and the max assignment fraction for each iteration.

[129] In Dynameq route choice is split into two steps referred to as Path Generation and Convergence Stage. For the Path Generation in the first iteration, all drivers choose the quickest path assuming that traffic flows at the free speed of each link. At the end of the simulation, the resulting link travel times are used on the next iteration. On the second iteration, for each assignment interval, half the drivers use the original shortest path and half use the new shortest path. This process continues, adding one new path at each iteration, until the maximum number of paths is reached. Thus, if 5 is specified as the maximum number of paths in the DTA specification, the first five iterations are used to find the five best paths for each O-D pair, for each assignment interval. In this case, on the fifth iteration, one fifth of the vehicles use each path for any given assignment interval. These iterations are the path generation stage of the DTA. During the remaining iterations (up to the number specified) is referred to as the convergence stage of the DTA, the number of vehicles using each path for each O-D pair and assignment interval is adjusted before each iteration in order to equilibrate the travel times. Two algorithms are available and are both based on the Method of Successive Averages (MSA). They are referred to as regular MSA and flow balancing MSA. When the path choices are such that the travel times on all paths are approximately the same within each assignment interval for each O-D pair, the network is said to be in a state of Dynamic User Equilibrium (DUE).

[130] Cube Avenue has two forms of route choice. For either option they use a list of paths generated for each iteration and segment up to a user specified maximum. The first option called "Packet Splitting" evenly distributed the number of vehicles in each packet among all available

shortest paths, thus creating new packets every time a new shortest path was found. The second option called "Packet allocation" never creates new packets but instead allows the existing packets to switch to a newly found best path with a probability inverse to the iteration number.

[131] Vissim uses two steps for its route choice procedure. The first is a "Path Search" where Vissim iteratively searches for the shortest path for each assignment interval. Multiple parameters are available to limit the number of available paths, reject paths who cost has increased too much, and more aggressively search for unused or comparably similar shortest paths. The second step is the "Path Selection" where two methods are available: Use Old Volumes, and Kirchhoff. Use old volumes does not look for a new path and is the equivalent of assigning all vehicles a previously determined path. The second method is distribution formula according to Kirchhoff and is based off a Logit model. The main differences is that the magnitude of the total difference between the difference of travel times of two paths and the total path travel time are taken into account (ie. The benefit of switching from a 20min path to a 10min path will be greater than that of a switch from a 130min to 120min path.)

[132] Pedestrian crosswalks can be modeled at any intersection with average hourly flows to account for pedestrian delays at intersections.

[133] Aimsun does have a plugin called "Legion" that is available under all licenses. Legion is a more in-depth pedestrian simulator but is only available in the microscopic modeling process.

[134] Pedestrian flows can be modeled in each intersection in terms of two-way crossing flows per hour.

[135] Up to 30 pedestrians can be modeled at once with the included Vissim license, otherwise an additional PTV product Viswalk is available for pedestrian simulation.

[136] TransModeler has three different ramp metering controls: Fixed Cycle (Pre-timed), Fixed Cycle (Actuated), Local Feedback (Closed-Loop).

[137] Aimsun has a total of 5 ramp metering schemes: Green-time metering (the cycle time and the green-time duration), Green-time by lane metering (the cycle time, the green-time duration and the lane offset), Flow metering (maximum allowed flow), Flow-ALINEA metering (initial flow, minimum flow, maximum flow, regulator parameter, calculation interval, desired downstream occupancy), Delay metering (mean delay for each vehicle and the standard deviation from the specified mean)

[138] DynuStudio uses a detector to control ramp meeting the three options available for the ramp metering detector: a user-defined, a feedback, or fixed. Based on the detector the ramp meter will release vehicles from a user specified start and end time (as many as needed) and between a user-defined minimum and maximum rate.

[139] Vissim uses the same menus that are used to program traffic signals to program ramp meters. Certain controllers (such as Vissig, a fixed time based controller) are included with all Vissim licenses. Others can be purchased as add-on modules or custom ramp meters can be programed via the API plugin.

6.7.1 Scripting and Application Programing Interface (API)

In the world of modern traffic simulation virtually nothing is “un-modellable.” Every software seen in this report has some type of scripting language or API that can be used to simulate something the software natively cannot accomplish. However, simulating something that is not natively available in the software’s library of tools is not trivial. For example, Aimsun does not natively have a toll or HOT lane pricing tool for operating such facilities. The Minnesota Traffic Observatory used Aimsun’s API to program in the MnPASS HOT lanes with their pricing algorithm even though Aimsun does not natively support it. This type of application takes months of an experienced programmer’s time so it must be taken into account when considering other programs that may have those features readily available.

6.7.2 TransModeler

TransModeler is a well-polished and capable traffic simulation suite. Out of all the selected software’s only one feature was not readily available and it is due to the way TransModeler is fundamentally designed. TransModeler is able to perform many of the advanced traffic management techniques sought after in today’s projects to deal with the ever increasing congestion and limited capacity available. It is also the only software out of the ones selected that can natively simulate reversible lanes inside a single object. Other software’s require the user to lay overlapping sections to produce the desired effect.

6.7.3 Aimsun

Aimsun came in second in terms of total number of software capabilities and features and many of the “No” features can be handled by another software such as Synchro. One feature of Aimsun that is not seen in any of the programs above but is definitively useful is the minimal file structure system. Besides the database and path file there is only one single file that holds all of an Aimsun network together. This reduced file structure makes it easier to share networks and be confident that no files were misplaced. Many of the other software’s have folder full of files that must be moved accordingly and associated with the program.

6.7.4 DynusT/DynuStudio

Out of all the software’s selected DynusT was one of the lowest in terms of requested features. Being an open source program any number of these features could be implemented but would require significant effort. Other drawbacks include the fact that individual lanes are not explicitly defined and therefore lane level phenomena cannot be replicated. The upside to all these is that given its course view of the network runtimes are shorter and with it being open source many research papers are available detailing its implementation into many different scenarios.

6.7.5 Dynameq

Dynameq represents the middle ground between all the software selected. Dynameq has the unique identity as being labeled mesoscopic but contains many microscopic properties. These microscopic properties allow effects, like lane usage, to be accurately simulated while still having the computational efficiency to model large areas. However, Dynameq has a much courser view of traffic signals and is only able to replicate single-ring controllers. Since there is no microscopic simulation software in the INRO suite of products another micro simulation suite would need to be used to model very detailed/complex networks.

6.7.6 Cube Avenue

Cube Avenue represents an interesting opportunity since the Regional Travel Demand Forecasting Model and the new Activity Based Model for the Twin Cities metropolitan area is based on Cube Voyager. Since Avenue is just an extension of Voyager the startup time could be greatly reduced. However as compared to the rest of the applications tested, it does have the least amount of features. Regardless, it can still be a viable candidate for adding a dynamic aspect to the RTDFM although restricted in the types of scenarios it could run.

6.7.7 Vissim

As was mentioned previously Vissim was added not because it was seen as a viable solution for a large Meso-DTA model but because it is used by many of the local stakeholders. The features that were desired in Vissim were looked for in other software more to show that Vissim was not the only option. Many of the stakeholder's comment on the fact that Vissim did not handle very large networks well so it is not considered a viable alternative.

6.8 SOFTWARE CONCLUSION

While all the researched software packages could potentially preform a DTA analysis of the Twin Cities, two stood out: Aimsun and TransModeler. Both of these software are not only feature rich but are able to perform multiple levels of simulation within the same software. This type of feature is extremely useful since it allows the user to perform large scale meso-simulations on the network while also being able to use the same geometry to perform microscopic analysis. Each of these models can also perform a Hybrid simulation with microsimulation on the complex portions of the network that could otherwise not be replicated by any other mesoscopic simulator. The most readily implementable DTA software would most likely be Cube Avenue since the network exists already, in the form of the RTDFM. Based on how Cube Avenue works though this would only be useful in studies based around major changes in capacity as it would not be able to examine the effects of more complex scenarios involving weaving, complex traffic control, or other lane dependent traffic phenomenon.

CHAPTER 7: DISCUSSION OF ALTERNATIVES

The initial, fundamental, question is if it makes sense to base the development of a large Meso-DTA simulation on the paradigm of the Regional Travel Demand Forecasting Model or on the approach followed by traditional traffic simulation projects. The difference is in the scope of the model and the structure allowing its reuse. The RTDFM[5] is a single model that has been developed, as a whole, through a considerable effort not driven by a particular project but by the need to have a stable resource for all and any planning and regional development program development and evaluation. Not only does it involve a single model but it also involves a comprehensive infrastructure of people and procedures for its maintenance and upkeep. Both the development of the model but even more the upkeep effort requires considerable funds. On the other hand, projects requiring the development of traffic simulation models, in general, each develops a new, limited scope, model for the purposes of the project. Such models are built on the most appropriate application, with the most appropriate resolution, and calibrated to cover the needs of the project. The cost of such models is much smaller than the RTDFM and although MnDOT keeps a library of them, the fact that they are split into at least two different applications and there is no active procedure of keeping them up to date, makes it very unlikely that a later reuse will save any effort. It is important to note here that it is not only the evolution of the road network that renders these models obsolete but also the advancement and changes made on the applications they are built upon. In most cases, newer versions of simulation applications do not even load old models let alone maintain their validity. The following sections expand on the pros and cons on each of the two alternatives.

7.1 SINGLE METRO-WIDE MODEL

By a single metro-wide model we imply that a specific Mesoscopic traffic simulation application is selected and upon that a model is built covering the entire Twin Cities metro region. The model might have similar coverage as the RTDFM but for mesoscopic traffic simulation instead of travel demand modeling. The framework for its maintenance and upkeep is part of the arguments. Since each point has positive and negative aspects pros and cons are mixed together.

1. Single resource: a metro-wide Meso-DTA model maintained as is the RTDFM would be a valuable resource for the transportation community. Similarly, with the RTDFM, all parties would base their projects on the same foundation and in most cases will directly use the model without having to add much to it beyond the hypothetical scenarios explored. This would reduce the effort involved on each individual project in terms of startup costs and extend the benefits from funding a single model.

[5] The RTDFM was last calibrated in 2000 using survey and speed data. It is now in the process of being replaced by the ABM that was released to consultants in the winter of 2016.

A requirement for this to take place is that this model needs to be built and maintained in a manner similar to the RTDFM. This implies a major investment for its creation and an even greater ongoing investment for the people and software infrastructure maintaining it. There have been very few instances where local consultants selected to maintain their own version of the RTDFM because the version provided by the Metropolitan Council did not cover their needs. Given the complexity of a traffic simulation model this need may increase invalidating the benefit received from the single resource.

2. Gradual improvement: the main resource can be updated as part of each individual project effort. The hypothetical scenario that is finally selected for implementation will be already integrated into the metro-wide model. The original investment will grow project by project, theoretically making each subsequent project easier and cheaper.

This implies though that there will be one individual, the model maintainer, who will be responsible for accepting or rejecting changes in order to maintain the integrity of the larger model. For example, on a large scale it is impractical to have all roads implemented. Like in the RTDFM, only the “important” links are included. If for a particular model a subarea is populated with all the links and that is returned to the metro-wide model, there can be path selection implications by having areas of the model with different link densities. The same can be said for traffic control. In short this is a pro only if the extra effort is spent in properly integrating each project’s work into the main model.

On the same subject it is also essential that each project that builds on the main model makes provisions for the modeling assumptions used to be included in the final report. Modelers make different assumptions or take shortcuts in order to make one particular part of the network work. If these assumptions are not known, it may backfire when others try to use a similar part of the main model. For example, DynusT is a probable software application that a single metro wide model can be based on. When modeling the effects of queue spreading DynusT cannot simulate vehicles in individual lanes on network links and if the queue extends from one link to another it would block the entire link instead of a single lane. This problem is most evident in cases where exit ramps are congested and the queue spreads onto the mainline. Normally this queue will not block the entire freeway but because of the way links are modeled in DynusT they will. Modelers often mitigate this issue by creating a separate link representing the maximum size of the queue on the freeway section. That link gets congested but the remaining freeway lanes are flowing since they are now on a separate link. Such a link cannot be too long since it impacts the capacity of the freeway section by dedicating that lane to only the exiting traffic. This method of mitigation can work for a particular level of demand and project since the modeler is aware of the assumption and monitors it to ensure it does not have unintended consequences. If another modeler were to now use the model, focusing on a different part of the network, and is unaware of this deviation from the truth it could generate problems with general model calibration.

3. Quality control: Keeping a metro-wide model active and updated means that the model can be run at any time with demands that match present conditions and check its accuracy and the accuracy of the

predictions made on prior projects. Also, having several people using the same model and exposing it to different inputs can uncover errors and bugs that have been missed. For example, it is very common in simulation that certain faults on specific locations, like traffic control bugs or improper priority rules on intersections, do not materialize unless congestion builds. During one project it is unlikely that the entire network can undergo a stress test but over several projects eventually the entire region will be scrutinized.

4. Interfaces and Extensions: A benefit from a single metro-wide model is that modelers will find it is important enough to develop extensions and interfaces for it. Normally for an individual project it would be infeasible both in terms of time and budget to develop applications for data manipulation and automatic importation. For example, the introduction of real measurements used in calibration is done manually although available programs offer scripting languages allowing the development of interfaces for their importation and direct use inside the application. Since each customer has this information in a different form it is unreasonable for the developer to produce such interfaces but if the investment for building a metro-wide model is made it makes sense to spend the additional funds to expand it over time and allow it to seamlessly interface with the local resources i.e. MnDOT loop detector database, MnDOT synchro arterial control database, the TomTom network speed statistics the metropolitan council has acquired, and other forms of traffic data. Extensions incorporating the MTO’s very own traffic control versions for ramp metering and HOT pricing algorithms are also good examples of extensions developed if a single model is maintained.

5. Information Database: the majority of traffic simulation applications currently in the market are based on a GIS platform. Each has its own but all allow the benefits of storing information relevant or irrelevant to the model. If a single metro-wide model is developed it can be used as the focal point for storing information about the network. Information on the location of loop detector on the freeway and arterials, location and control parameters of traffic lights or other types of traffic control (stop or yield signs), ramp metering controllers, and other can all be integrated and stored in the same GIS DB and be seamlessly tied to the Meso-DTA model.

There is one caveat to this. Depending on the application, such a use may inflate the file size of the model and/or increase the necessary computing resources needed to even open it. It is perfectly reasonable to include in the investment the purchase of a computer big enough to nullify this issue but it could be unreasonable to expect that every consultant or potential user of the single model can make such an investment. A good example from the history of traffic simulation is the Transims simulator. In that case it was the complexity of the model structure that went all the way defining every little detail of the transportation experience of an individual. Very thorough application able to create extremely robust models of large areas with the only caveat that only one computer in existence can ever run them, the supercomputer at the Argon national laboratory in Illinois.

6. Development and upkeep cost: although this issue was alluded to several times already it is important to highlight that although the cost of developing a single reusable metro-wide meso-DTA model is

considerable, the ongoing cost of maintaining the human and organizational infrastructure necessary for its maintenance and upkeep is even more considerable. While the initial cost could be reduced by building it gradually, the upkeep cost cannot be avoided if all the benefits described earlier are the goal. As discussed later in this document, Caliper determined that the development of a Meso-DTA model is a feasible task with a reasonable cost and timeframe. In their estimate, they did not include any particular project work like work zone staging, capacity improvements, etc. These would need to be modeled by in house staff. MTO students worked for almost three years to produce a working model of the Twin Cities in Aimsun. While this effort does not compare directly to a practicing engineers effort it does represent the amount of manual labor need.

7. Locked to a specific traffic simulation application: the RTDFM of the Twin Cities metropolitan region was originally build in TRANPLAN. When TP+ was retired as a program, Cube Voyager was its replacement. The metro council didn't really have a choice in this switch since the cost of building the RTDFM in, for example, EMME/2 would have been as big or bigger as the original investment. The same way now, the new Activity Based Model development again made sense to stick to Cube Voyager since it considerably reduces the cost. The same way a single metro-wide Meso-DTA model will be tied to a particular application making it nearly impossible to use or switch to another one without a considerable amount of investment. Although the Travel Demand Models (TDMs) have more or less matured as applications, mesoscopic traffic simulation applications are still very new and are still evolving. This evolution implies that not all features necessary for different projects are available or good enough on a single application. This will probably change as these applications mature but it could be too early to place a bet and make some projects impossible to use the resource.

8. Larger potential benefits- while a small project might not have a budget large enough to justify the building of a Meso-DTA model, decision-making might still benefit from mesoscopic analysis. A regional model would allow the benefits of Meso-scale analysis to be realized on a larger group of projects.

7.2 SMALLER PROJECT-SIZE MODELS

Logically the discussion regarding the choice of not developing a single metro-wide model but to focus on the development of methodologies and requirements for the development of different, project-based, models follows most of the arguments already made in the previous section. From the stakeholder interviews an almost unanimous opinion was that the freedom of using the most relevant application for each project and the freedom to model each particular project's subject piece of the Twin Cities network are so important that a single model is not practical or desired. On the other hand, it is interesting that the same people stressed the need for building a library of models that can be reused or combined in later projects. These two desires, though as it was stressed already, are mutually exclusive. Experience has shown that the reusability of a model is virtually nonexistent if not maintained even just one year after the end of a project. In that short time the evolution of the software alone invalidates most of the work done. For example, multiple stakeholders interviewed in the earlier stages of this project talked about projects in Vissim that did not translate correctly between versions and thus changes had to be made to the network to get consistent results. The MTO has personal experience with Aimsun in which during the transition from version 7 to 8 multiple features changed and lane changing models were enhanced to the point where old calibration values caused gridlock and thus had to be changed to default values and calibration needed to be revisited. Additionally, without the proper model documentation structure it is much more efficient for another modeler to redo the work rather than try to understand what his/her predecessor did. Regardless, the following are discussions regarding the pros and cons of having project bounded Meso-DTA models.

1. Project result accuracy: Developing a model specifically for a particular project almost always guarantees that the results produced are the best for answering that project's questions. A lot of the steps that guarantee those results cannot be generalized or implemented on a bigger model. For example, during the recent exercise of developing a Meso-DTA model[6] for deciding the staging of the next three to four-year construction projects in the west part of the Twin Cities metro, the consultant followed a well-established procedure of gradually building the network necessary for the project. A large piece of the Twin Cities metro region was extracted from the RTDFM and imported into DynusT, specific assumptions regarding the arterial control were made sacrificing some of the model robustness there for the sake of producing credible results on the freeways. Having acquired a credible model of the large area, a subarea was extracted generating the necessary traversal matrices. This subarea was further manipulated to improve the models accuracy and match with present data. Without delving into that project in greater detail it is assumed that these steps produced a good base for answering the project's questions. Unfortunately, none of these steps are easily reversible and produced a model that cannot be integrated into a bigger one without a significant level of effort. The reason the calibration was made on the subarea is because there were not enough resources to do it on the regional size model and it was not logical to expect that after these changes are introduced in the bigger model, that model will continue to match present conditions. If it could, it would have invalidated the need to work on a subarea. So, this is a good example of a project driven by specific needs, producing credible results, but leaving behind a resource that would save very little effort on a future project. From the methodology followed, which we assume was necessary, it is clear that if the desire was to keep the model re-usable this accuracy level would have not been possible. An interesting point on the subject of this particular project is that although a DynusT model of the Twin Cities had been developed approximately two years earlier as part of the 35W bridge collapse, the current consultant opted in building this project's model again from scratch instead of re-using the earlier model. We assume this decision was made because this course of action was at least more economical if not the only one viable.

[6] The DynusT Meso-DTA model was in development and actively being used during the entire length of this project. All relevant details obtained are listed in a later section.

2. Tool matching the job: when a specific project is considered, the first requirement is to choose the right tool for the job and available resources. The selection of DynusT in the west metro Meso-DTA construction staging study as the application used was the result of a discussion that considered several other applications most of them excluded because the project resources and timetable did not allow their perceived requirements in terms of detail and data. Similarly, different applications offer features that are currently unique or are uniquely implemented. It is not reasonable to expect that any single application can cover all possible needs. Having the freedom to choose the right application for the job was one of the stakeholder requirements. This reduces greatly the reusability of a model but it can be necessary given the project requirements. For example, if the project requires the modeling of the MnDOT ramp metering algorithm, DynusT would not be an application used without major software development.

3. User comfort and expertise: as it is good to have the right tool for the job it is also good to have the right modeler for the job. Different consultants have invested in different applications and they have developed considerable expertise and tools for them. Locking down all projects to the same applications would considerably impact fairness and competition in the local market. Now projects are awarded based on merit as well as budget. The same can even apply in the context of a single firm. Different engineers on the same firm have formed expertise on different applications based on the most specific feature they are good at.

4. Scope growth: this project was born from MnDOT's need to answer questions involving parts of the network too big to handle through microscopic simulation and too complicated to use the RTDFM. This need did not materialize now but only recently the software and hardware computing resources became available to even contemplate traffic simulation as a tool. This evolution will continue and it is reasonable to believe that the scope of the projects will also grow only because it will be possible to accomplish them. As with the recent example of the DynusT model, sizes will increase and it can end up that there will be projects covering the entire metro. If a project oriented model development is followed, the effort in entering all the necessary data will only serve the particular project and leave no lasting benefit. Additionally, this method of project specific models makes it highly unlikely that a really comprehensive and detailed model of the Twin Cities network or part of it would ever be produced or compiled.

CHAPTER 8: STEPS AND GOOD PRACTICE FOR MESO DTA MODELS IN THE TWIN CITIES

As was previously mentioned there is a lot of experience in the use of the TDMs and how to calibrate them. TDMs often incorporate a very large geographical area that is computationally difficult and nearly impossible to calibrate without considerable effort. Furthermore, the objective of most TDMs is to forecast traffic volumes, project travel demand throughout the entire geographical region covered based on changes in land use and demographics, and not to simulate the effects of those changes on traffic at the operational level since it is not validated to that level of detail. The same can also be said for microscopic models. Both of these model types are firmly established in the transportation modeling community with a lot of experience and past projects to learn from. The newest type of mesoscopic simulation or even large microscopic simulation models do not have well established steps for constructing a model. This section is designed to establish those steps or good practices for building a large scale Meso-DTA model.

8.1 COLLECTION AND ASSEMBLY OF EXISTING DATA

Many types of data should be obtained in order to develop a large Meso-DTA model. Many of these are available but will require significant effort to collect and consolidate into a consistent format. Some of the items that should be assembled include:

- Aerial imagery
- Signal timings
- Arterial traffic count data
- Freeway traffic count data
- Speed data

8.1.1 Aerial Imagery

Aerial imagery covering the entire study area should be obtained through a source and at a level where individual lane markings are decipherable which generally occurs around 1-meter resolution but in some cases, such as freeways, larger resolutions may be acceptable. In addition, the photos should be geographically-referenced to reduce the effort needed to import them as a background layer. Many simulation packages have licenses to use aerial imagery from sources such as Google, or Bing. ArcGIS, a popular GIS software, also has free aerial imagery sets included that were found to be acceptable for creating simulation models. However, if a major change in geometry has occurred it may not be available through the free sources and thus a paid service would be needed such as DigitalGlobe[7]. The primary role of the aerial imagery from any source is to determine number of lanes, intersections (including left/right turn bays), and interchange geometry.

8.1.2 Signal Timing Data

An important aspect of a Meso-DTA model and a key feature missing in most static models is signals at junctions. Often signals are the cause of significant delay and these should be accounted for in any large Meso-DTA model. Even though this additional detail would add additional realism it has significant drawbacks that would have to be overcome.

- Acquisition – One of the most difficult aspects of programing signal timings for a large geographical area that spans many different municipalities is acquiring them. At this time, there is no database of all the signals in the twin cities. To collect these signal timings effort needs to go into contacting each municipality that has control of them. These could include city, county of state agencies and if the Meso-DTA model were to include the entire Twin Cities Metropolitan area there would be 200+ cities, 16 counties and 2 state agencies from which to obtain traffic signal information. In addition, many of the signal controllers are not network connected so in order to collect the most current signal timings an engineer would have to locally visit each site. Efforts are being made to network connect major signals and unify them with a single database. MnDOT recently began implementing, in the Fall of 2016, a new "Centralized Traffic Signal Control Software" (CTSCS[8]) to control nearly a third of its signals with the goal of connecting all signals to it. This type of consolidation will greatly reduce the amount of effort required to acquire signal timing data.
- Format – As was brought up during the interview process there are many different signal controllers located in the region ranging from advanced digital controllers to old fashion mechanical timers. Excluding the rapidly declining mechanical timers, it was revealed that some cabinets do not even interpret inputs the same as others. An example of this occurs in the interpretation of yellow time, some controller's use a percentage of red time to define yellow time is included in the red time while others need it explicitly defined. These types of difficulties can vary in scope and often the engineer needs to read each controller's individual manual to interpret the timings.

[7] The UMN is fortunate to have an agreement with DigitalGlobe that allows us to connect to their base maps through any GIS software. It was found that aerial imagery is both clear (sub 1 meter) and in some cases very current (less than 6 months); https://www.digitalglobe.com/

[8] http://www.dot.state.mn.us/its/projects/2011-2015/systemsengineeringforctscs.html

The MTO over the years has acquired signal data from multiple municipalities and has seen many different types of format. They range from synchro files with no time of day listed, to large database files full of generic split times. Most modern signal controllers can output a full spec sheet for a given intersection which will allow you to fully model that given signal but these require a significant amount of manual effort in order to process. To successfully implement signal timings on a large scale for use in a Meso-DTA model it is imperative that some universal format be established. This would allow signal plans to be entered into a database and simplify the process of importation. The benefits of a database are that it could not only contain current/past signal timings but also contact information so that collecting the most current timings would be streamlined.

- Synchro - Most of the traffic signals in the metro region have been developed or are part of a Synchro model at some point given its widespread use in signal retiming projects. Therefore, it is imperative that for any Meso-DTA network to be successful a link with Synchro is necessary. This link could take the form of being able to directly import a Synchro files (".syn" files) or being able to process the Synchro Universal Traffic Data Format (UTDF[9]) files.

8.1.3 Freeway Traffic Count Data

Freeway data for the twin cities is easily obtained through the use of MnDOT's traffic management and database system IRIS.[10] The data can be directly obtained through a separate servlet[11] to handle data requests for the protected traffic data database. This data includes volume, occupancy, flow, headway, density, and speed and is available for all freeway locations in intervals as small as 30 seconds. The only drawback to this data is that it is not regularly cleaned so construction impacts, detector malfunction, or other forms of corruption are not always recorded. Therefore, when using the freeway detectors, it is usually the practice to take several days outside the construction season (October/November) to help balance the data.

8.1.4 Arterial Traffic Count Data

Given the vast size of a Meso-DTA model and the amount of arterials included it is imperative to include traffic counts on them. A traffic count data collection plan should be developed with the objective of capturing, to the extent possible, the pattern of traffic flow and distribution through the Twin Cities

[9] Universal Traffic Data Format (UTDF) is a standard specification for transferring data between various software packages.

[10] http://iris.dot.state.mn.us/

[11] http://data.dot.state.mn.us/

regions of interest favoring heavily used routes and bottleneck locations such as bridges over major freeways and rivers. Data collection sites should also be collected so that count data would be available between all major communities along the primary corridors in order to allow for the calibration to more accurately reflect traffic demand patterns entering and leaving those communities.

These should include both tube and turning movement counts (TMC) in the smallest intervals possible with 15 minutes being preferred in order to capture the effects of peak spreading and to calibrate the model to localized issues. The counts should be obtained from the most recent years on record that have data throughout the entire day. However other more course data, such as AADT's or peak only TMC's, while not preferable is better than having no data. Given the unavoidable reality that this course data is often all that is available and calibrating to this traffic count data from different days, years and or seasons while not ideal, would only be useful if they could be broken down into more refined counts based on localized traffic patterns. In the case of AADT's if an entire 24-hour demand could be modeled by the simulator then they could be more or less used as is. With sparse and wildly varying traffic counts comes the understanding that the errors in the goodness-of-fit will be larger relative to smaller simulation studies. Again, the most useful data for a Meso-DTA model would be traffic volumes/speeds in 15 intervals or less.

Another potential source of future data is whether to included data from signal controllers. Currently many signalized intersections use detectors (magnetic loops, vision, or radar) to actuate the lights. As was proven in the development of the SMART[12] signal system it is possible to record these individual activations and save them to a database. This data is invaluable as it would provide modelers with high-resolution data over years at location that typically require expensive data collection procedures and only provides a single isolated day. Given the new Centralized Traffic Signal Control Software (CTSCS) that MnDOT is implementing it is potentially feasible that the detector information at signals controlled by the CTSCS could be stored within the already established IRIS database. Additionally, MnDOT already has 70+ Automatic Traffic Recorders (ATR) that continuously record traffic volumes to produce state wide AADT's. These should be unified into a database similar to IRIS and data saved in as small intervals as is feasible.

8.1.5 Speed Data

In 2010 the Metropolitan Council obtained average speed data from TomTom[13] for use in developing the ABM model and are associated with it's link network. Therefore, if in a given model there is a reference or index to the ABM network, the TomTom speed data could be directly associated to certain links.

[12] SMART was developed at the UMN in order to collect high resolution traffic data to generate real time performance messures. http://www.its.umn.edu/Research/FeaturedStudies/smartsignals/

[13] http://www.tomtom.com/lib/img/HISTORICAL_TRAFFIC_WHITEPAPER.pdf

While this is the most freely available form of speed data (not including the freeways) in the Twin Cities it is quickly becoming outdated and would likely not be purchased sooner than every 5 years if at all. Vendors such as HERE[14] (Formally Navteq), INRIX[15], TomTom, or AirSage[16] can provide historical speeds, travel times, and traffic patterns for nearly every street in the metro area. For routes that are of particular interest at certain times of the day and aggregated data is unavailable or sparsely available individual speed runs using GPS should be performed to give the complete picture. All of this speed data is used to identify issues where speed may be effected by non-congestion related issues such as road geometry, steep grades, tight turns, small sight distances, etc.

8.1.6 Stop/yield sign database

In addition to signalized intersections, un-signalized intersections also produce significant amounts of delay in a DTA model. Every stop, yield, or significant traffic control sign needs to be modeled in order to apply the appropriate amount of delay at each intersection. These types of features can be time consuming to implement into a model and could be incorporated into a separate database.

8.1.7 Congestion location/queue lengths

Another set of data that helps to visually validate the model is the use of congestion maps and queue lengths. Every year MnDOT produces a freeway congestion report of the metropolitan freeway system where congestion is defined as sustained speed under 45mph using the freeway detectors described previously. Most of the same companies that supply the speed data previously described also can generate these maps using their data. These maps help to locate sections of the model that should show signs of congestion and vice versa. Though rarely collect queue lengths can also help to visually see if bottlenecks are causing queues to spill far beyond where they do in reality.

8.1.8 Saturation Flow Data

In general, in all forms of traffic simulation, saturation flow information is needed. Especially in congested networks, where long queues are developed on the arterial network, it is necessary to capture the right queue discharge headways. All traffic simulation models have specific parameters that control this behavior and, although the default may be reasonable in a lot of cases, road geometry, elevation, sight distances and other factors affect this behavior. Additionally, in areas where

[14] https://here.com/en

[15] http://inrix.com/industry/public-sector/

[16] Developer of software that aggregates signaling data from cellular networks to provide real-time speed and travel times for major roads. http://www.airsage.com/

considerable pedestrian activity is observed, the effect on the saturation flow rate especially on right and left turns is important to be captured. Simulation applications that distinguish lanes in a section also include some parameter or combination of parameters that can affect saturation flow rates per turning movement. Although pedestrians and other traffic calming measures are not necessary to be included in the model, their effect can be replicated by adjusting these parameters.

8.1.9 Truck percentages

Another big calibration issue is to verify the truck demand is adequate. If too many trucks are simulated, you may have an issue where too many trucks are diverting to alternate routes and causing unreasonable amounts of congestion on the arterial network. Therefore, truck percentages are needed on the major freeways and arterials to not only verify the truck demand is valid but also to ensure the impact of trucks is correctly simulated in the model. If truck percentages are unavailable proper steps should be taken to avoid invalid truck routes.

8.1.10 Travel Time Data

While some would not consider travel time data essential for model calibration it is an excellent tool for validating the model and identifying large discrepancies to ensure that the model is able to match more than one set of data. For example, if the travel time simulated through a corridor was much greater than the validation data it could indicate that unrealistic congestion is forming, signals are miss-timed, or speeding is occurring in the real data. On the other hand, if your travel times are much shorter than reality it might indicate that there is not enough congestion or other factors like pedestrian crossing are impacting travel time. Travel time data can be acquired through many different sources. They could be manually collected via GPS using probe vehicles or through the use of an Automatic Traffic Recorder (ATR) that collect data such as Bluetooth MAC address[17] in order to calculate travel times. Both of these options have significant equipment costs associated with them and require manpower for installation of the ATR's or driving probe vehicles. Commercially available products would appear to be the best solution and thorough the previously mentioned vendors (TomTom, HERE, etc.) travel times can be collected.

8.1.11 Origin-Destination Survey Data

If time and funding permits a further validation of the model can be conducted on the OD matrices through the use of Origin-Destination survey data. This data could be collected using the ATR described in the section above or through the use of a license plate survey. Both of these options are known to be cost prohibitive and commercially available products could potentially provide the same data at a

[17] http://roads.maryland.gov/OPR_Research/MD-12-SP909B4D-Bluetooth-Traffic-Detectors_Report.pdf

comparable cost. The OD data would be used to ensure that the model was able to represent not just the volumes at the major gateways to the study area, which can be done using the traffic count data described previously, but also the volume of trips passing entirely through the study area and where they enter and exit is correct. This data can also ensure that the route choice algorithm in the simulation suite is performing correctly and choosing the appropriate routes on major freeways.

8.1.12 Database

Based on the required data needs of a large Meso-DTA model it would appear that a database of all this information should be designed. Having a centralized database would not only reduce the effort of collecting this data but would allow all agencies in the region a uniform location to enter data and reduce overlapping collection. It would also serve to identify where additional data would need to be collected.

A tool that appears to meet these needs is already commercially available and implemented by the city of Minneapolis. The software, MS2[18], is a cloud based Transportation Data Management Software package that uses an internet GIS based software to store and visualize all data contained within it. The software can provide a database not only for turning movement counts but also traffic signals, pavement markings, road signs and travel time/HERE data. The last one being significant since it could complement the turning movement counts to in most cases provided complete network coverage for the calibration and validation of a Meso-DTA model.

8.2 GEOMETRY

Information about roadway geometry and roadway characteristics is needed to develop an accurate lane-level model of the road network. While in some simulation suites individual lanes are not represented they still must be specified in order to have accurate values for capacity and other roadway characteristics that are directly impacted by the number of lanes. One of the most valuable sources for this information is the GIS file produced by the RTDFM model. The RTDFM GIS network contains most of the major roads in the Twin Cities and already has data such as the 2010 TomTom speed data associated to it. The only downfall is that in some cases the RTDFM has made assumptions in the network that effect the centerline location of the roadway. An example comes from the consolidation of one-way pairs into single two-way streets. While this works for the RTDFM’s macroscopic simulation method it would not be accurately depicted in a mesoscopic model. Therefore, the most industry accepted and efficient way of checking geometry is to directly derive it on top of aerial imagery. The MTO did attempt to use the different Open Street map importers available in several of the simulation software suites. It was found that as of the time of writing the amount of effort required to check/fix the network after

[18] http://www.ms2soft.com/

importation was as much and often more effort than if the network had been built from scratch atop aerial imagery.

8.3 CONTROL

Signal timing data could be considered the one of the most complicated and time consuming data entry process of a Meso-DTA model. These timings are often in many different formats and are not readily importable to a software suite until significant work has been made to decipher them into a universal format for importation. It is often found in practice that for very large models it is infeasible both in time and resources to collect accurate signal timings throughout the entire network. Therefore, many of these large models use default signal timings or a pattern/style of signal timings to give the network some sort of control information. Some programs even recognize this difficulty and provide ways to synthesize traffic signal timings based on simulation results in order to reduce delay. For smaller networks, signal delay as a whole may represent a large portion of the overall delay and thus must be modeled accurately and worth the additional effort need for importation.

Even if signals exist, problems can still occur during importation. For example, many municipalities use Synchro to optimize traffic signals and many software suites have built in tools to import directly from Synchro files. When the simulation is run for the first time, each signal must be checked to ensure that there are no errors and if there are, they must be manually corrected. If a signal is missing, additional effort would have to be applied in order to generate a signal for that intersection. Given that in many networks there are a various number of signal timings through the day this process of manual checks can take a significant amount of effort. Some software suites also do not support all the different styles of traffic signal controllers. DynusT, for example, cannot replicate dual ring controllers which are common throughout the Twin Cities. In order to model dual rings they would need to be converted into single ring controllers which would introduce both error and additional work for the modeler in order to implement them.

Another important signal timing issue that must be addressed in either large or small networks is the implementation of ramp meters. In the Twin Cities ramp meters are deployed at nearly every ramp and use a complex metering algorithm to maintain a peak flow on the main line. It was found through an independent study[19] that shutting down the ramp meters caused a 9% reduction in freeway volume and a 22% increase in freeway travel time. If ramp meters are not programmed into the model, then a similar result should be expected from the simulator over the real data.

[19] http://www.dot.state.mn.us/rampmeter/study.html

8.4 DEMAND

A regional TDM is a crucial building block of a Meso-DTA simulation model. For the simulation model to be a valuable tool for future planning studies, it relies on a travel demand model to produce estimates and forecasts of daily traffic demand. The simulation model uses these generated traffic demands to analyze traffic impacts for different project scenarios or can be used to loop back with the TDM to improve the estimates and forecasts. Either way a link with the local TDM is needed for any large project to provide useful results. For the Twin Cities this could be the RTDFM model but given that the RTDFM model is being retired for a more current TDM, the Activity Based Model (ABM), it would be inefficient to use the RTDFM as a base.

At the time of this report the ABM was going through the final stages of testing and was released to local consultants. Geometrically the model changed very little so previously developed methods to import the GIS network or to cut subareas remain valid. The biggest changes that would effect a Meso-DTA model are the increased number of Traffic Analysis Zones (TAZ's) and 30-minute passenger car demand intervals. The increased TAZ from 1632 in the RTDFM to 3360 in the ABM increase the number of in/out points that can be used by vehicles. The biggest change comes from the 30-min demand intervals for passenger cars. This refined demand interval will allow the Meso-DTA model to load the network more realistically then the hour long demand intervals of the RTDFM.

It should be noted that there are other forms of demand that could be used such as turning movement counts or a third party vendor such as AirSage. These would not be as useful since the major use of a large Meso-DTA model is to predict traffic impacts on a large scale for future projects. This need inherently requires a TDM so that future demand can be based off of multiple factors instead of just a percentage growth of current demand which is what a service such as AirSage provides. Additionally, both of these methods would be cost prohibitive. Turning movement counts would require a massive amount of manual labor to deploy traffic counters to collect the data and would also require additional effort to synthesis the demand. While AirSage data prices increase significantly as more TAZ's are added that would consume an erroneous portion of a projects budget.

8.5 CALIBRATION/VALIDATION

Historical traffic data, as outlined in the previous section, represent the principal source of information for calibrating the model. The specifics on how each model can be calibrated vary greatly between the different simulation suites due to the different assumptions made in each one in terms of traffic flow theory and route choice. However, two main calibration techniques that are developing headway in the modeling community are discussed in the following sub sections.

8.5.1 Origin/Destination Matrix Estimation (ODME)

As is being seen in most recent large scale Meso-DTA models a method that is often being used to calibrate these large models is known by several different names but most commonly as Origin Destination Matrix Estimation (ODME). Traffic counts are used to estimate origin-destination matrices, sometimes referred to as trip tables, or adjust a previous one to more accurately reflect the traffic counts provided. Thus, the ODME seeks to drive down the error between simulated and observed volumes. Most methods of matrix estimation employ some form of Static traffic assignment (Static-ODME) with the most advanced methods also using simulation (Dynamic-ODME). A virtue of the simulation-based approach to ODME is that it is capable of producing, in addition to numbers of trips between OD pairs over the simulated period, a temporal distribution of departures and the spatial distribution of trip ends around the origin and destination traffic analysis zones (TAZs). In other words, not only the magnitude, but the complexion of the demand, in terms of departure time and departure and arrival location within TAZs, that best agrees with the time-varying count data emerges from the process. The temporal distribution of trips is preserved in the form of time-varying trip matrices, which are derived from the table of trips and their departure times. Trips usually are aggregated by 15-minute departure interval each time the ODME was completed.

There are various well understood shortcomings of origin-destination matrix estimation methods and these limitations are summarized below:

1. Traffic counts cannot reveal either the origins and destinations of vehicles or their routes.
2. Poor coverage in a study area may leave links/sections on key routes between origin-destination pairs without counts, and will in turn degrade the quality of the ODME solution.
3. Modelers are tempted to combine counts from different days, years, and/or seasons to increase coverage. They also often combine counts together to reflect an “average” day, yielding counts that when combined into a single set do not represent any observed reality.
4. There are multiple possible ODME solutions, and if there have been major changes in time, demographics, or networks the solution may not represent a correct origin-destination matrix despite matching counts.

These limitations are sources of error, uncertainty, and inconsistency in the ODME solution. It is thus important to understand these limitations and design a plan in order to minimize them when using ODME. The poor coverage limitation can be a major source of error and discrepancy. Preferably every path generated should pass through at least one count station for the ODME procedure to be the most effective given that any path not captured between an OD pair cannot be adjusted. An example of this would be shorter paths that may not pass through a count location, such as a freeway detector, and would make capturing every path infeasible. Given the nature of ODME procedures there is no concrete requirement for the number of count location needed to be successful. Many of these requirements are based off the engineer’s judgment while developing a model. A rule of thumb for a successful ODME is that there are enough volume collection locations in the network so that at least 80%[20] of all paths between all OD pairs contain at least one such location. This implies that a lot of volume counts are required on the arterial network. This is where they are the rarest and the most disparate in terms of time and condition of collection. If the count locations are not homogenously spread over the entire network there is the fear that the model is validated not because the error is minimized but because it is disproportionally transferred to areas with no data.

Additionally, these ODME matrices will not hold true for all time. Just as Travel Demand Models must periodically obtain "ground truth" a Meso-DTA models ODME procedure must be re-run periodically with updated traffic volumes in order to account for factors not seen in the correction factors over the period of several years. Previously the Metropolitan Council had updated its Travel Behavior Inventory (TBI) every 10 years with the last one being in 2010. Currently however, the Met Council has acquired funding[21] to perform home interview surveys along with other TBI elements every 2 years. This data would then be used to refresh the ABM approximately every 5 years. Given that the Meso-DTA model will likely be based off the OD's from the ABM it should be assumed that the ODME must be re-run after each new TBI is incorporated into the ABM. This could result in a significant level of effort on the modelers part. In a proof of concept model described in Section 9 of this report a single Dynamic-ODME run for the network took 3 weeks for a relatively coarse network of the Twin Cities. A more detailed network could take anywhere from 2-3 months of computing time based on the developers estimates and this level of additional effort must be taken in to account when relying on a ODME procedures to calibrate matrices.

8.5.2 Simulation-based Dynamic Traffic Assignment

In order for reasonable route choices to be simulated, congested, or loaded, travel times on such routes must be estimated. This is the primary function of the simulation-based dynamic traffic assignment (DTA) stage in the methodology. A full simulation is executed iteratively, with the method of successive averages (MSA) applied to output travel times on each iteration. The route choices of each run are thus a function of the travel times simulated and averaged over prior runs. Dynamic, 15-minute travel times are estimated using the simulation-based DTA. Through this dynamic assignment, dynamic, 15-minute travel times (and the dynamic route choices) are expected to stabilize (i.e., drivers cannot switch to paths they perceive to be better). The averaging of the travel times is intended to "smooth" the travel times over multiple iterations to prevent inefficient and counter-productive "flip-flop" between good

[20] Pg.125, E. Bert, A.-G. Dumont and E. Chung (Dirs.). *Dynamic urban origin-destination matrix estimation*

[21] Currently funding is available for the next 6 years however, a framework for establishing funding to collect TBI data on an ongoing basis is in progress.

and bad routes from one iteration to the next. The assignment runs until it converges to a target User Equilibrium relative gap, the same measure used in the traffic assignment stage in the TDM, or until a maximum number of iterations is reached. Unlike the traffic assignment in the TDM, however, the relative gap is not generally relied on as the stopping criterion in the application of the simulation-based DTA. Because the traffic simulation model is a stochastic Monte Carlo simulation (i.e., each simulation is initiated with a different random seed and will produce variable results) and because vehicle trips are discrete (i.e., they cannot be divided into tiny fractions as they are in the static traffic assignment methods), relative gaps of the order of magnitude expected of static traffic assignments in the TDM cannot be achieved. Empirical studies have shown that simulation-based assignment methods cannot do better than 2-3%, or a relative gap of 0.02 or 0.03.

Meso-DTA models can be run on various build and no-build scenarios to evaluate the performance of alternative or proposed projects. A Meso-DTA model can be run, depending on the software, either from a cold start, where drivers usually assume free flow conditions in the first iteration, or from a warm start, in which the solution of a previous Meso-DTA informs the route choice decisions of drivers in the first iteration. A cold-start DTA must be run for a greater number of iterations since no previous solutions are known and a lot of trial and error must occur in order to build a history. Depending on the software this usually takes anywhere from 50 -100 iterations to converge when the DTA is run from a cold start. The number of iterations can be greatly reduced in some cases when a warm start is used and only slight network changes have occurred. When significant changes to the network are made though, for instance to simulate the impacts of multiple construction projects, a cold start is generally advised.

Given the size of the network there are several different techniques one can employ to improve the accuracy and run time of a DTA simulation. While a cold start, as discussed above, involves starting the network empty with no knowledge of shortest routes, a warm start considers a list of shortest routes or at least a list of preferred routes. In many cases it is very difficult to perform a cold start for the entire demand since the number of iterations required for the model to converge are very high. In such cases a gradual load of the network can be performed. For example, one can reduce demand uniformly to less than 50% of the normal. This will almost guarantee an uncongested network and allow the cold start to rank routes with the natural shortest ones between O/D pairs high on the list. Having obtained such routes additional demand is introduced in subsequent simulation circles but instead of providing a shortest path tree based on free flow conditions the path tree produced from the previous run is used. This method reduces the search of the MSA algorithm and allows the DTA to converge faster.

8.6 SUGGESTED OUTPUT

While the following list does not represent every output necessary from a DTA model, since many of these will be project dependent, it does highlight several outputs that may not be readily available to output. Some of these outputs require additional effort from the modeler ahead of time to ensure the simulation model is capable of producing them and to make sure they are output with each simulation run. Many simulation suites by default turn off most of the output in order to decrease the time for the simulation to finish.

- Road User Cost – The roadway user cost is extremely important for the purpose of analyzing a work zones impact on the network and determining when, where, and how to stage them. It is also used to determine if special equipment, such as movable barriers, or other forms of mitigation can be justified by the additional savings in road user cost through their implementation.
- Delay – Delay goes hand in hand with road user cost since it can help to identify issues associated with high user costs. There are cases where your delay per vehicle may only be a few minutes but because the road has such a high volume the road user cost is high and may appear to warrant advance mitigation strategies when in reality you may only achieve a few minutes' reduction in delay.
- Queue Length – Queue length is another important measure in determining the impact of congestion on the network and how far from the point of bottleneck the queue extends. This is important to know if bottlenecks caused by construction, ramps, or other features extend and consequently affect the travelers on a different section.
- Travel Time by Mode– Like delay travel time is important to see the change in travel time due to some change in the network. Separating it out by mode such as single occupancy vehicles (SOV), high occupancy vehicles (HOV), buses, light rail, etc. can help to determine if a bottle neck impacts all vehicle modes and help predict if ridership on other forms of travel would increase.
- Reliability of System – In order to have faith in a model's results some form of checking the reliability of the system was indicated as being useful. Clients indicated that the model should arrive at similar results when single non-important links/sections are altered. For example, changing the speed of a minor arterial section should not have an impact on a freeway 5 miles away.
- Vehicle Miles Traveled (VMT) – VMT is often used to determine the environmental impacts of bottlenecks and construction. This could also be used as one of the reliability measures previously stated to ensure that the model runs are comparable as a large increase or decrease in VMT between comparable model runs could indicate an underlining issue.
- Speed/Density/Flow/Travel Time by Link – As was indicated by the clients there are often times where there is so much data requested that static maps and tables can get overcrowded and hard to interpret. Interest was expressed in using a Graphical User Interface (GUI) or Geographic Information System (GIS) to display link attributes individually. This type of architecture helps to complement the table of values by seeing those values visually on a map and allows the client to look at individual sections to see the impact of the particular modeled scenario. The only downside to this is that all the information needs to be packaged into a movable and easy to use format so that it can be used by others without needing extensive training or specialized software.

8.7 FUNCTIONAL REQUIREMENTS

This section lists a brief list of functional/computational requirements that should be followed in order to achieve a successful and useful Meso-DTA model.

8.7.1 Run time

In order for the model to be useable the run time must be kept to a minimum. Having a model that takes several days to run makes calibration an issue since the user would have to wait several days to see if the change implemented worked. Issues also arise when there is no active logging on multiday runs of current progress to see if a gridlock condition has caused the simulation to break and produce useless data. Through the interviews it was generally seen that 16 hours would be an upper cutoff for simulation run time. This allowed changes to be made and have it run overnight and be ready in the morning when the modeler returns.

8.7.2 Computational Requirements

The computational requirements of a simulation model happen to be the most quickly changing requirement of them all. Given the pace at which computers are constantly increasing in power and decreasing in price makes it difficult to establish a base level. However, based off the interviews the computer should be able to run from a modeler's office and not need a "supercomputer" to run. For comparisons sake the MTO as part of this project will be using two computers, Table 2, to test the different simulation packages. When told these specifications and prices every interviewee agreed that the base computer was easily obtainable or a comparable system possessed already. None of the interviewees had a computer like the second one but said that if it was needed it could most likely be obtained.

Table 2: MTO Test Computer Specifications

Attribute	***First (Base) Computer***	***Second Computer***
Processor Name	Intel Core i7 2600k	Dual intel Xeon E5 2643 v2
Total Processor Specs	4 cores (8 threads) @ 3.4GHz	12 cores (24 threads) @ 3.5GHz
RAM	32 GB DDR3 (4x8GB)	256 GB DDR3 (8x32GB)

System Type	64 bit	64 bit
Operating System (OS)[22]	Windows 7	Windows 10
Graphics	GeForce GTX 560	GeForce GTX 750
Hard Drives	120GB SSD (OS) 500GB WD Black (Database)	240GB SSD (OS) 1000GB WD Black (Database)
Price	~$1500	~$10,000

8.7.3 Activity Based Model Integration

Integration with the Metropolitan Council Activity Based Model (ABM)[23] is seen as essential since it will more the likely be the main source for OD information for the Meso-DTA model. While there were only a few instances of using demand data not obtained from the RTDFM they were almost always too costly or complicated to use continuously over time. Therefore, it is essential that the DTA model can input demand data obtained from a maintained source such as the RTDFM. However, the RTDFM at this time is undergoing a major update to change from a traditional four step model to an activity based model. With this new model comes a large update to the number of zones (1600+ to 3200+) from which origins and destinations are generated. As this new activity based model will be the future forecasting model for the twin cities it is important that any Meso-DTA model is able to import the demand data from it with little effort required by the modeler.

8.7.4 Model Maintenance

In order for the Meso-DTA model to be used for multiple projects in the future it's would be required that some agency house the model and collectively improves and distributes the model as needed. They would also serve as trainers to guide individual firms in using the model to ensure the work they do can be used to improve the model over time. This centralized system would allow the model to be used over

[22] Windows 7 has an architectural limit of 192GB of RAM

[23] The ABM was released to consultants in the winter of 2016

time and reduce the startup cost of having to build a model from scratch allowing more scenarios to be run and ultimately providing quicker turnaround times between when a client asks a question and receives an answer.

CHAPTER 9: TRANSMODELER MODEL FEASIBILITY/EFFORT NEEDED

Before introducing this section's findings from the experiment with TransModeler, it is important to revisit why TransModeler was singled out versus other software packages. One of the main reasons TransModeler was selected was due to its feature rich package and relatively cheap licensing cost when compared to other DTA simulation packages. In section 7 a detailed breakdown of 6 traffic simulation suites; TransModeler, Aimsun, DynusT/DynuStudio, Dynameq, CUBE Avenue, and Vissim was provided. Unfortunately, due to the need for MnDOT to have a Meso-DTA model to plan the 2017 construction year, the development of such a model could not wait for the findings of this project. The consultant hired by MnDOT chose to use DynusT/DynuStudio to build the Meso-DTA model and was able to deliver MnDOT the results they needed to pick their construction staging plan for the 2017 calendar year. At the time of writing the model was undergoing additional calibration. The other reason why TransModeler was chosen over others was due to the fact that the developers at Caliper have offered to produce a reasonable "rough" network at no cost. This includes creating tools that would automate the importation of MnDOT detector data, attaching centroids to the network, and starting to calibrate the freeways to better match count data from MnDOT's Intelligent Roadway Information System (IRIS).

9.1 DYNUST MESO-DTA CONSTRUCTION PHASING MODEL

The DynusT/DynuStudio model developed for MnDOT is still undergoing calibration and thus some details of the model are unknown and could not be tested given that the MTO does not maintain a license for the DynuStudio software. All data about this model was collected from a draft report dated 3-21-16 as well as the DynusT files dated 11-06-15. Based on that report it is assumed that a regional model is still infeasible to run in terms of the project timeline and man-hours available for calibration.

9.1.1 Software

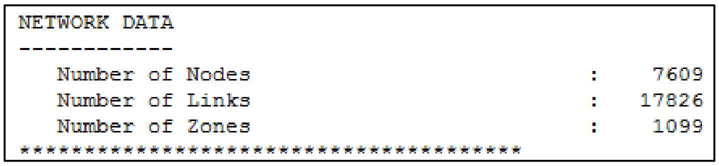
Before proceeding into a breakdown of the Construction Phasing Model it is important for the reader to know the difference between DynusT and DynuStudio. DynusT is an open-source simulation-based Dynamic Traffic Assignment (DTA) software developed originally by Dr. Yi-Chang Chiu at the University of Arizona. DynusT does not natively have a Graphical User Interface (GUI), however a free version of a GUI called NEXTA[24] is distributed with it. Due to the open source nature NEXTA is not being developed at the same pace as other traffic simulations software's with the last release in Jan 2014.

[24] Developed by Professor Xuesong Zhou at the University of Utah

DynuStudio on the other hand is an attempt to construct a Commercial GUI interface based on DynusT. It is part of RST International which was founded by Dr. Robert S Tung who is also continuing to develop DynuStudio. DynuStudio also contains additional tools for generating reports and automation tools for generating all of the files needed for DynusT to run. DynusT networks created through DynuStudio are not easy to operate without it and, although comprised mostly of text files, these files are not easily interpreted due to the lack of headers or column labels.

9.1.2 Network

Figure 5 below shows the extent of the DynusT regional model that was imported directly from the Metropolitan Councils Regional Planning Model and used to extract subarea traversal matrices in the outlined region. Figure 6 shows this outlined region in more detail. The subarea consists of 7609 nodes (928 signalized), 17826 links, and 1099 zones.



```
NETWORK DATA
------------
   Number of Nodes                          :    7609
   Number of Links                          :   17826
   Number of Zones                          :    1099
***************************************
```

Figure 4: DynusT Network Meso-DTA model summary

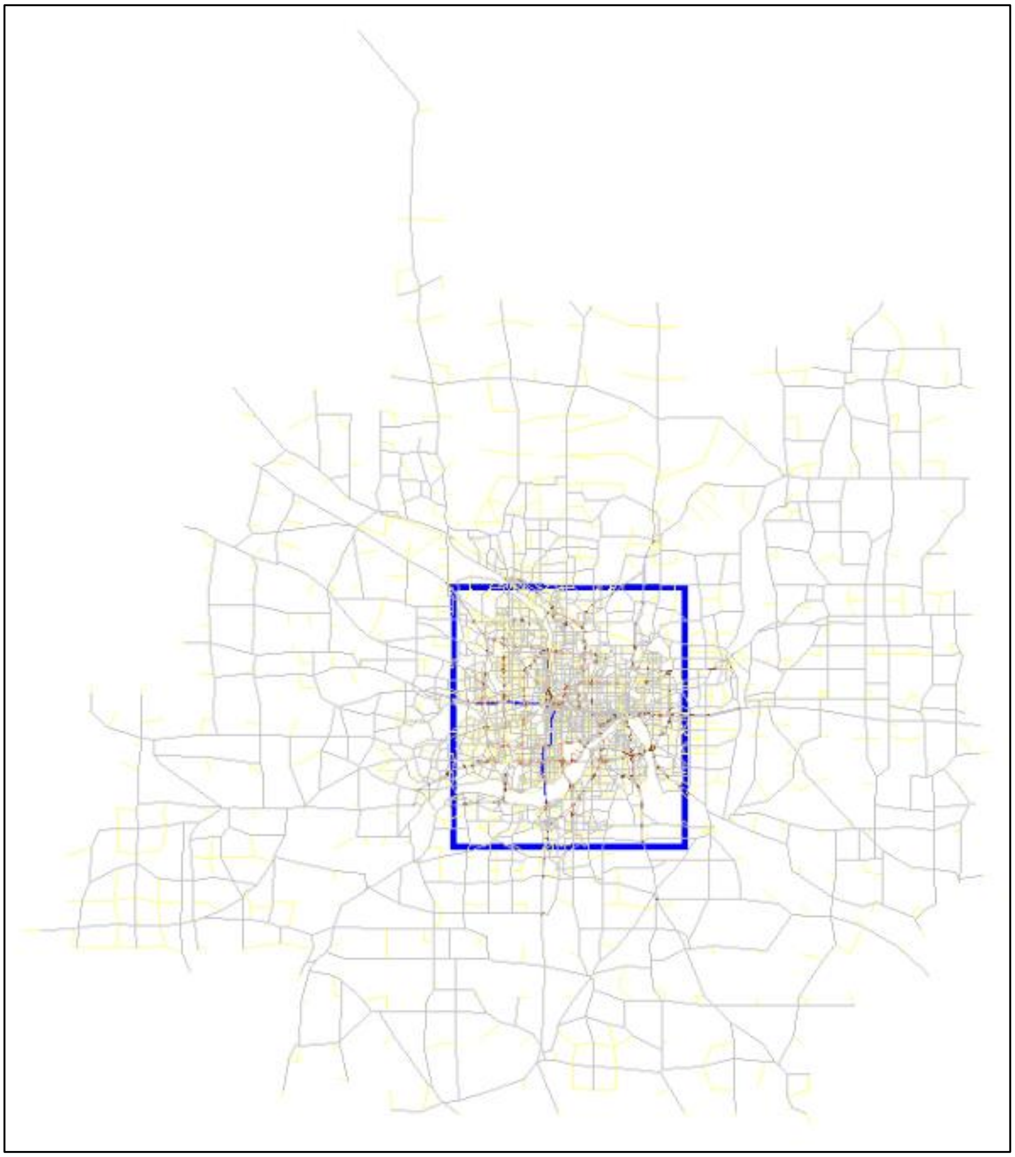


Figure 5: DynusT Regional Model with Subarea Outlined

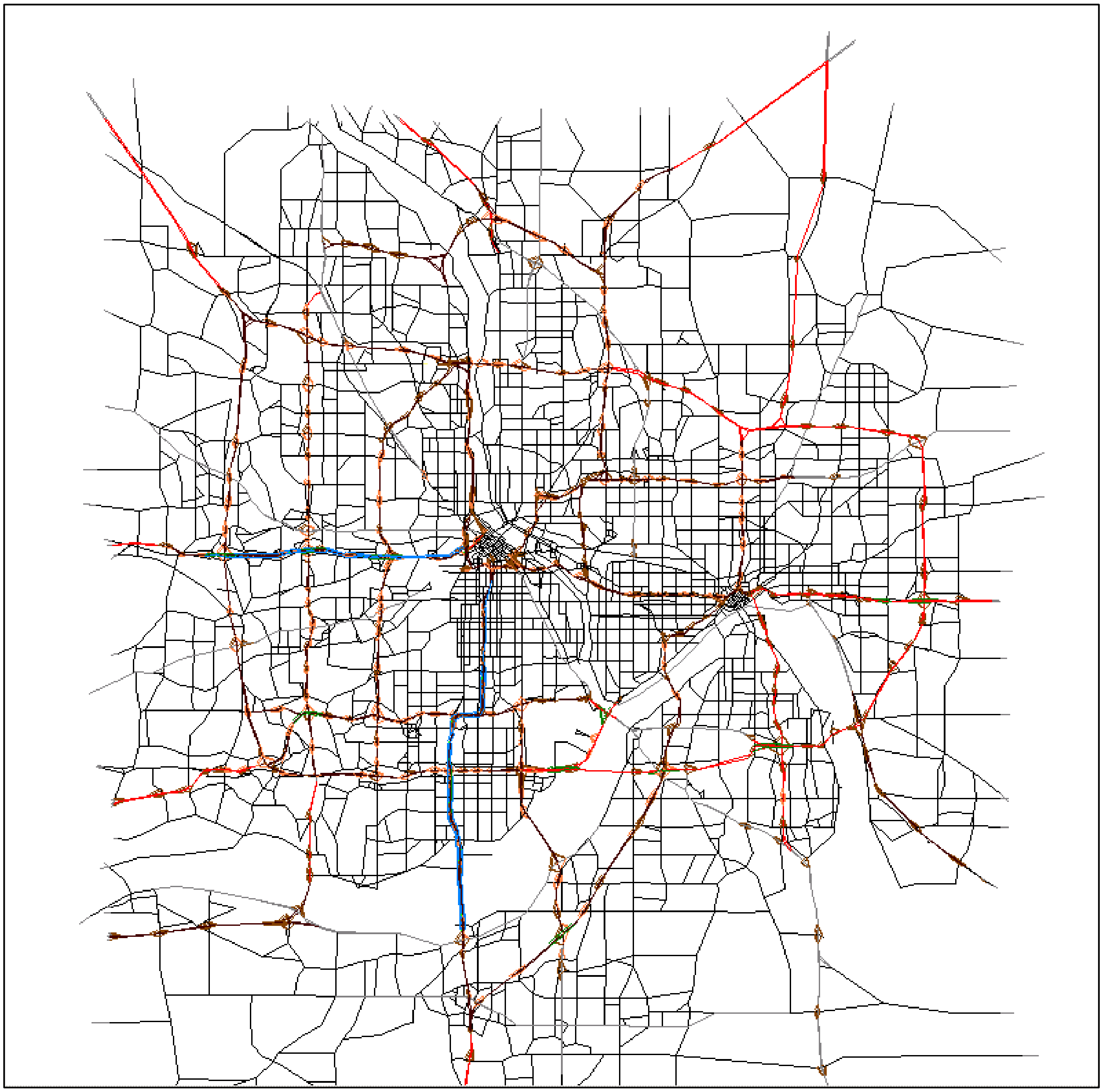
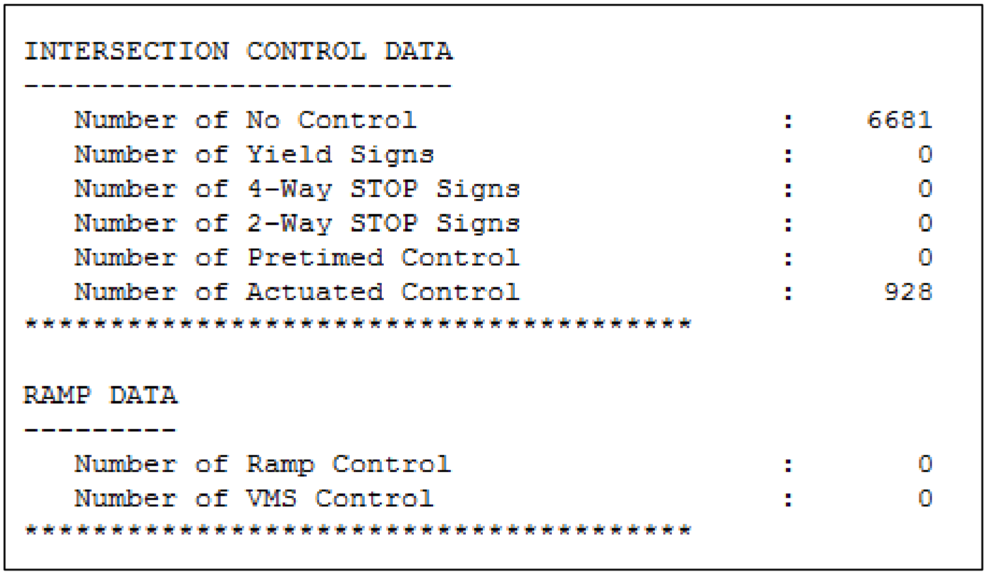


Figure 6: DynusT Subarea Model

9.1.3 Control

In total 6,681 intersections are contained within the DynusT network as seen in Figure 7. Of the 6,681 intersections 928 intersections were singled out as having an actuated control plan. For these 928 signals the consultants used a built in tool in DynuStudio to calculate the best control plan per intersection. The tool is called “Calculate Default Phasing & Timing (3+ phase with protected left).” This tool used a completed 24-hour demand run with no signals as a base to determine the best signal plans. All other intersections had no control information listed. Additionally, no ramp meters were programed into the network.



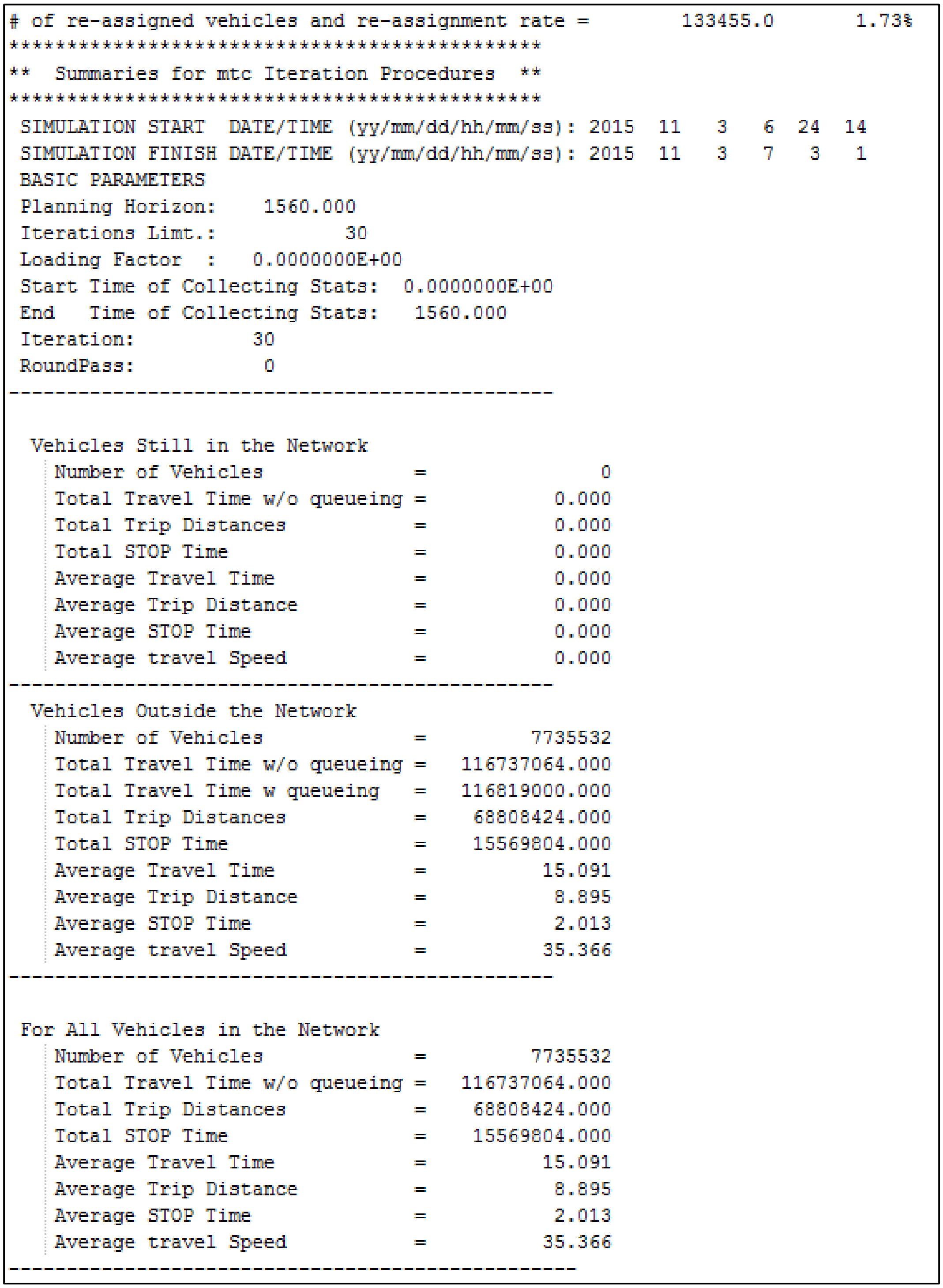
```
INTERSECTION CONTROL DATA
-------------------------
   Number of No Control                       :      6681
   Number of Yield Signs                      :         0
   Number of 4-Way STOP Signs                 :         0
   Number of 2-Way STOP Signs                 :         0
   Number of Pretimed Control                 :         0
   Number of Actuated Control                 :       928
***************************************

RAMP DATA
---------
   Number of Ramp Control                     :         0
   Number of VMS Control                      :         0
***************************************
```

Figure 7: DynusT Meso-DTA model control info

9.1.4 Demand & Simulation info

In total 7,376,706 Cars, 332,471 Truck, 26,355 HOV Vehicles were simulated over a period of 26 hours (24 hours of demand + 2 hour cool down). The simulation interval or frequency at which vehicle details are updated as they make their way through the network was set to 6 seconds. This simulation interval is the only time when vehicles can be generated or changes in vehicle details such as speed can be made. DynusT has two additional sub intervals called the assignment interval, when paths from every Origin-Destination (OD) are recalculated, and aggregation interval (statistics calculation) which are both based on the simulation interval. Given that both were set to 50, new paths are produced and statistics being calculated every 5 minutes. Overall this was done for 30 iterations and achieved a relative gap (R-Gap) of 5.9% and a reassignment rate of 1.73% (133,455 vehicles) as seen in the Convergence.dat and outMUC.dat respectively[25]. These two numbers represent how close the model is to achieving convergence. The R-Gap measures the average difference in travel time between every OD pair and the shortest path available (per aggregation interval) while the reassignment rate is a measure of how many vehicle paths are being changed between each DTA run in order to reduce the R-Gap. This whole process took around 20 hours for all 30 iterations. Figure 8 shows the final iteration of the subarea with statistics on travel time, Stop time, and travel speed for all vehicles.

[25] An error was found in the “MnDOT DTA Model Draft Documentation” in which the writer mistakenly interpreted the re-assignment rate found in theOutMUC.dat file as the relative gap.



```
# of re-assigned vehicles and re-assignment rate =        133455.0       1.73%
**********************************************
**  Summaries for mtc Iteration Procedures  **
**********************************************
 SIMULATION START  DATE/TIME (yy/mm/dd/hh/mm/ss): 2015  11   3   6  24  14
 SIMULATION FINISH DATE/TIME (yy/mm/dd/hh/mm/ss): 2015  11   3   7   3   1
 BASIC PARAMETERS
 Planning Horizon:    1560.000
 Iterations Limt.:           30
 Loading Factor  :   0.0000000E+00
 Start Time of Collecting Stats:  0.0000000E+00
 End   Time of Collecting Stats:   1560.000
 Iteration:          30
 RoundPass:           0
-----------------------------------------------


  Vehicles Still in the Network
    Number of Vehicles             =              0
    Total Travel Time w/o queueing =          0.000
    Total Trip Distances           =          0.000
    Total STOP Time                =          0.000
    Average Travel Time            =          0.000
    Average Trip Distance          =          0.000
    Average STOP Time              =          0.000
    Average travel Speed           =          0.000
-----------------------------------------------
  Vehicles Outside the Network
    Number of Vehicles             =        7735532
    Total Travel Time w/o queueing =   116737064.000
    Total Travel Time w queueing   =   116819000.000
    Total Trip Distances           =    68808424.000
    Total STOP Time                =    15569804.000
    Average Travel Time            =         15.091
    Average Trip Distance          =          8.895
    Average STOP Time              =          2.013
    Average travel Speed           =         35.366
-----------------------------------------------


 For All Vehicles in the Network
    Number of Vehicles             =        7735532
    Total Travel Time w/o queueing =   116737064.000
    Total Trip Distances           =    68808424.000
    Total STOP Time                =    15569804.000
    Average Travel Time            =         15.091
    Average Trip Distance          =          8.895
    Average STOP Time              =          2.013
    Average travel Speed           =         35.366
-------------------------------------------------
```

Figure 8: DynusT subarea model final iteration

9.2 TRANSMODELER - TWIN CITIES MODEL

As was previously discussed, and with the TAP's approval, the final task of this project was to evaluate the level of effort needed to make a working TransModeler model. The developers at Caliper, have put forth considerable effort to produce a rough network. This section will outline where the model is at this time and the additional effort that is needed to bring it to the level of detail seen in the DynusT model.

9.2.1 Network

While importing the network from the regional model would have been preferred, Caliper suggested that, given the time frame, it would be easier for them to build it from scratch than to import the regional demand models stick network and fix the errors caused during importation. The downside to this fact is that Caliper would be limited on the amount of network they could recreate. Even so with approximately 160 man-hours[26] of effort they were able to produce the network seen in Figure 9 with the level of detail seen in Figure 10.

[26] Caliper did not explicitly log hours for this network. The overall effort from network creation to calibration was estimated at 160 man-hours.

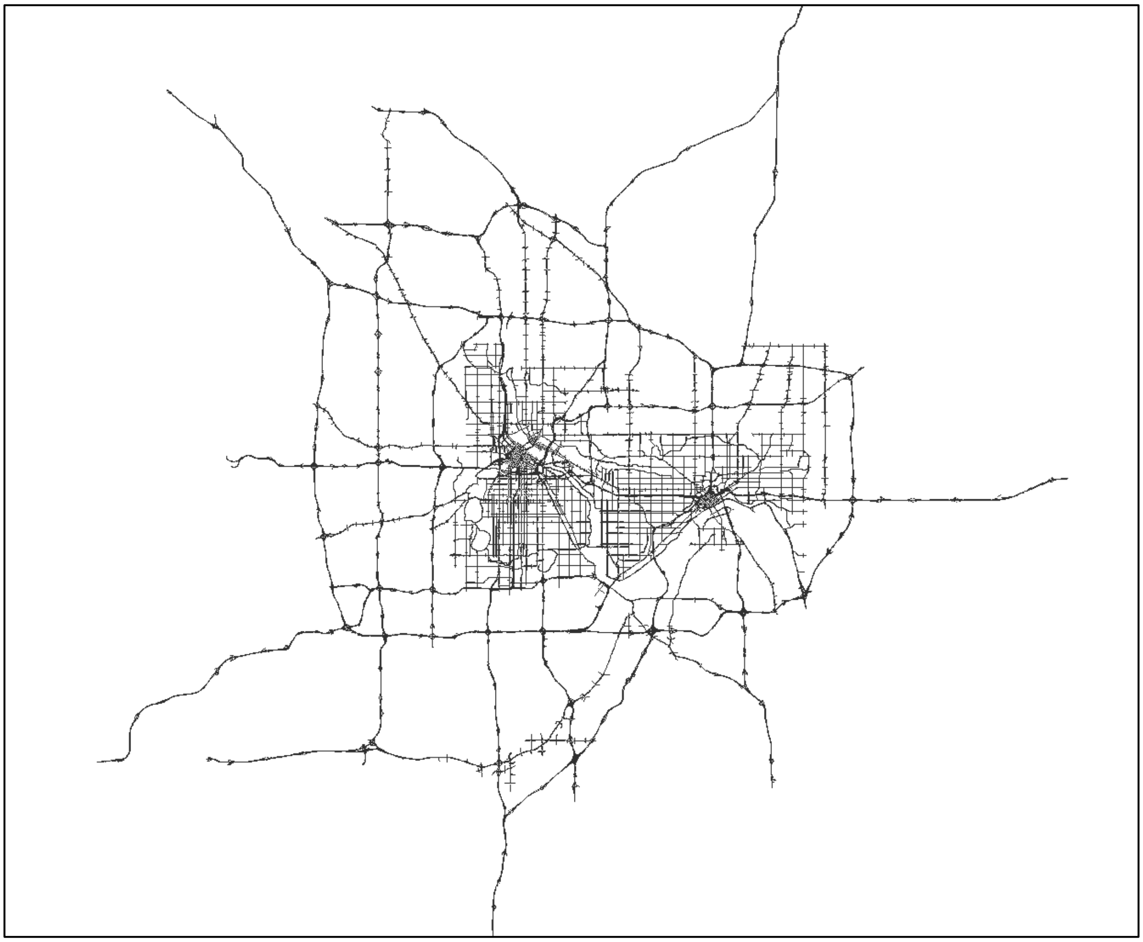


Figure 9: TransModeler TC model full extent

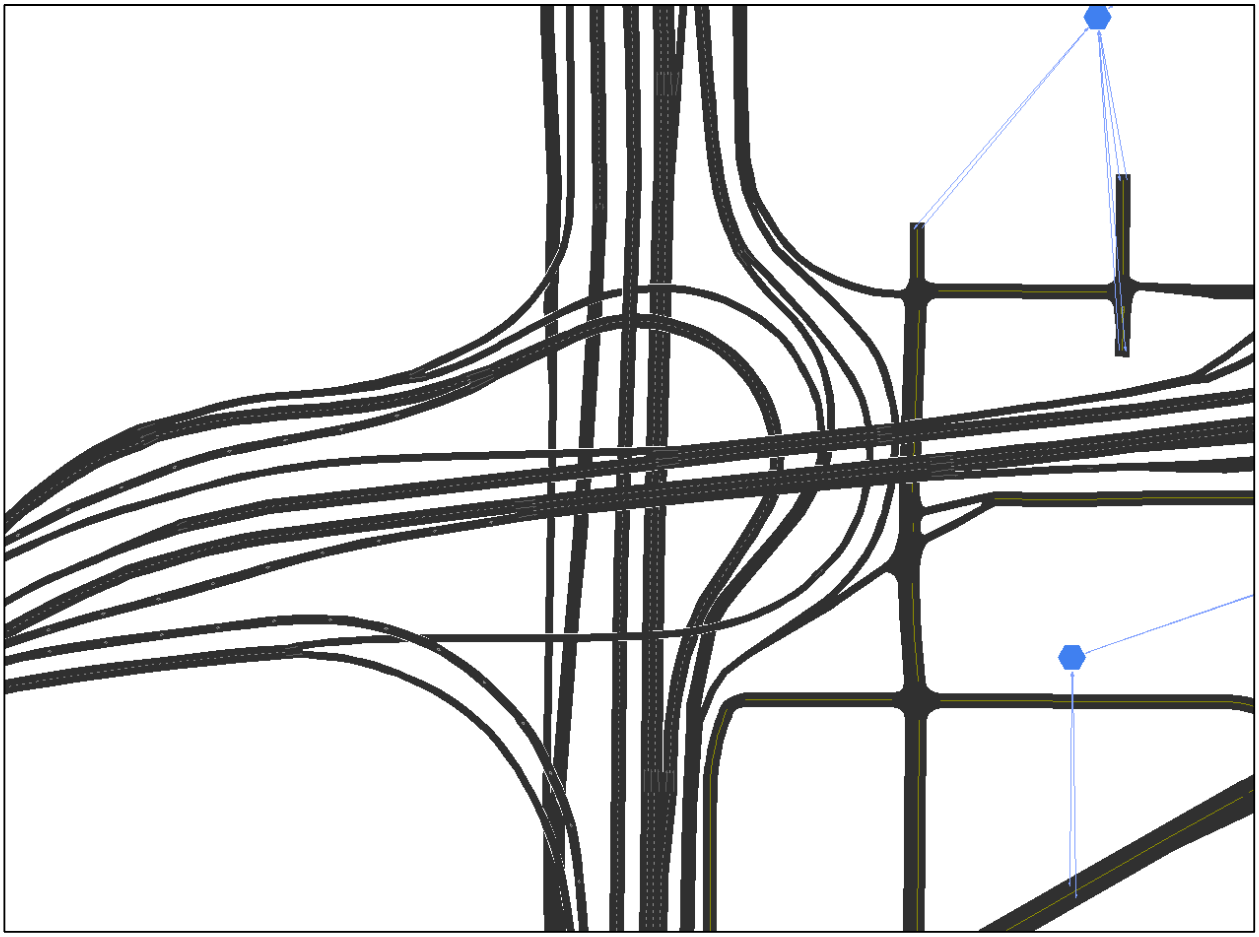


Figure 10: TransModeler TC model geometric detail - I-394/I-94 interchange

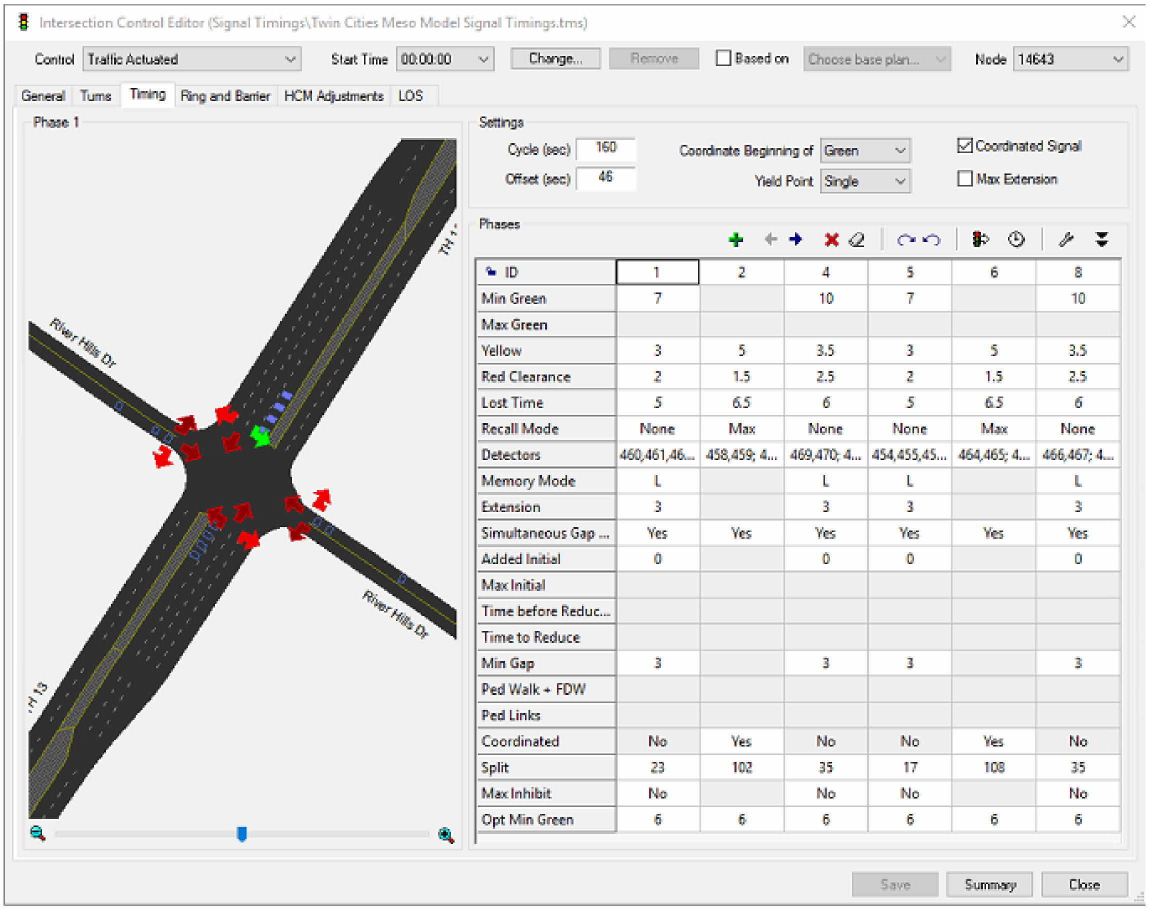
Since Caliper was going to run their Dynamic Origin Destination Matrix Tool (ODME) tool they opted for doing only the major freeways outside the Minneapolis/ St. Paul city limits since freeway counts from IRIS were readily available and the ODME tool does not function properly when sparse counts are available. As discussed in section 9.5.1, it would be preferable to have every path between every OD pair pass through at least one count station. This could be true if every driver used at least one section of the freeway network, however many trips begin and end using only the arterial network. Based on this fact and that there is no concrete requirement for the number of count location needed for an ODME tool to be successful, the number of count locations is based on the engineer’s judgment while developing a model. Some researchers have found that 80% of all traffic flows need to be intercepted in order to obtain a satisfactory result[27].

Based on the experience of the research team during a training session held at Caliper headquarters, the effort required to add additional geometry would be minimal. For example, during this project Caliper sent over an initial network for us to look over to see if there were any major routes that should be included but were not in the model yet. The team identified 6 corridors that should be added to the network and asked the Caliper team to add them to the network. In total the additional network edits consisted of approximately 38 miles of roadway and around 120 intersections. The team identified these corridors on a Friday and the request to include them was sent out at 3:00pm that day. The researchers received an updated network with all the suggested revisions the following Tuesday at 7:00am. Assuming no one worked on the weekend all of these additions were made within one business day. This fast turnaround in combination with the researcher's personal modeling experience in TransModeler would indicate that adding in all the roads in the Metropolitan Councils Activity Based Model contained within the 694-494 ring road would be less than a months' effort for an experienced user.

9.2.2 Control

One of the hardest parts of simulating arterials is acquiring and coding-in signal timings and was discussed in section 9.1.2 in detail. As part of this task the research team obtained all of MnDOT's synchro files in an attempt to have them available to import into the network. The developers at Caliper looked at several of the Synchro files and determined that given their current state it would take a significant amount of effort (1+ months) to go through all 161 networks. One of the main issues with the networks is that they are not geo-located and were developed by many different modelers and thus are not consistent. Caliper determined that each of the Synchro files would need to be imported into TransModeler and then exported to geo-locate them and convert them into a consistent format. Depending on the size of the Synchro network this can take around 30 min to 1 hour each. Caliper did however, as a test, import two corridors totaling 39 intersections from Synchro files for the AM peak, one of Hwy 13 and the other of Hwy 51 as seen in Figure 11 and Figure 12.

[27] Pg.125, E. Bert, A.-G. Dumont and E. Chung (Dirs.). *Dynamic urban origin-destination matrix estimation*



Control: Traffic Actuated
Start Time: 00:00:00
□ Based on
Node: 14643

Settings

Cycle (sec): 160
Offset (sec): 46
Coordinate Beginning of: Green
Yield Point: Single
☑ Coordinated Signal
□ Max Extension

Phases

ID	1	2	4	5	6	8
Min Green	7		10	7		10
Max Green						
Yellow	3	5	3.5	3	5	3.5
Red Clearance	2	1.5	2.5	2	1.5	2.5
Lost Time	5	6.5	6	5	6.5	6
Recall Mode	None	Max	None	None	Max	None
Detectors	460,461,46...	458,459; 4...	469,470; 4...	454,455,45...	464,465; 4...	466,467; 4...
Memory Mode	L		L	L		L
Extension	3		3	3		3
Simultaneous Gap ...	Yes	Yes	Yes	Yes	Yes	Yes
Added Initial	0		0	0		0
Max Initial						
Time before Reduc...						
Time to Reduce						
Min Gap	3		3	3		3
Ped Walk + FDW						
Ped Links						
Coordinated	No	Yes	No	No	Yes	No
Split	23	102	35	17	108	35
Max Inhibit	No		No	No		No
Opt Min Green	6	6	6	6	6	6

Figure 11: TransModeler Signal GUI example of Hwy 13

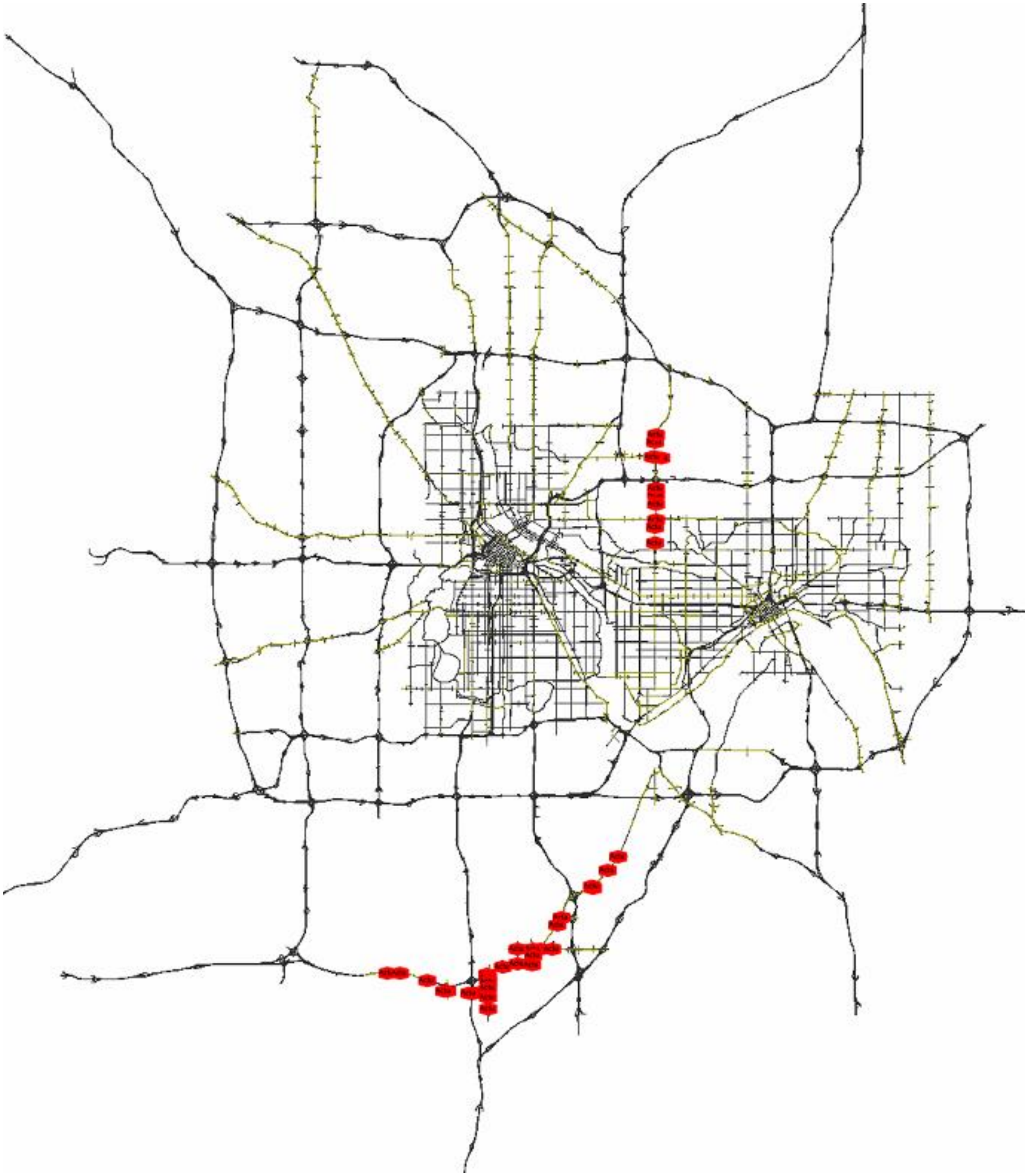


Figure 12: Twin Cities TransModeler model with signalized intersections imported from Synchro Highlighted

However, while discussing this issue with MnDOT personnel, it was discovered that MnDOT is in the process of moving to a new Centralized Traffic Signal Control Software (CTSCS) that will potentially be able to export signals in a more consistent format and thus reduce the amount of effort needed. MnDOT would prefer that all of its signals be in a single system and is actively pursuing an architecture that will allow other agencies to have their signals contained within the system as well. This system is also quoted as having the potential to log arterial count data that would help with the calibration of the model and to assist the ODME procedure. As was previously mentioned this system is being implemented during the final months of the project and will not be operational in time to be evaluated. Additional research is warranted to validate these claims and to develop tools to export this valuable data.

Given this information the research team and Caliper decided to focus on loading the network with demand instead of attempting to load in all of the traffic control. In addition, while not in the current version of TransModeler, a tool has been developed by Caliper to use the turning movements from previous simulation runs to optimize a set of signals based of a default ring-barrier template. This allows the user to implement a vast number of signals in batch in order to have some control in the network. This is the same/similar technique used by the consultants while developing the DynusT subarea model in DynuStudio. This method of implementing traffic signals is exponentially more efficient than importing Synchro networks. Until the new tool is released the research team cannot quantify the exact time savings associated with it.

9.2.3 Demand

Caliper was able to introduce some demand into the network using the trip based Regional Travel Demand Forecasting Model. Due to the custom cut size, it was easier for Caliper to extract demand matrices from their own program, TransCAD, instead of CUBE where the RTDFM natively runs. Once loaded into TransCAD they were able to extract subarea matrices for the network previously seen in Figure 9. These matrices would be used as a seed matrix for the Dynamic ODME tool, which is an in-house tool that will eventually be included in the software, to produce more reasonable OD matrices before some limited calibration. However, in order to run the Dynamic ODME, volume data was needed in as many segments as possible. Upon providing Caliper with a raw XML file from MnDOT's freeway management system, IRIS, they were able to write a script that associated the GPS coordinates from the rough locations of the detectors in IRIS to the TransModeler network. This script was able to match a total of 998 valid count locations which they associated with 15-minute aggregated volume data that the MTO team extracted from September/October of 2015. The output of the Dynamic ODME, after around 3 weeks of computing effort, resulted in 4 matrices, Single Occupancy Vehicles (SOV), High Occupancy Vehicles (HOV2, HOV3) and Trucks, for the AM Peak Period of 6AM-9AM with totals seen in Table 3Table 3: TransModeler Total Demand after Dynamic ODME.

Table 3: TransModeler Total Demand after Dynamic ODME

DEPARTURE TIME	SOV	HOV1	HOV2	TRUCKS	TOTAL
6:00 AM	60,109	7,697	7,032	5,790	**80,628**
6:15 AM	69,962	8,387	7,667	6,727	**92,743**
6:30 AM	73,985	8,695	7,852	6,791	**97,323**
6:45 AM	74,112	8,693	7,839	6,810	**97,454**
7:00 AM	75,261	8,855	8,013	7,266	**99,395**
7:15 AM	77,954	8,984	8,074	7,303	**102,315**
7:30 AM	77,839	8,920	8,081	7,132	**101,972**
7:45 AM	76,290	8,830	7,809	7,132	**100,061**
8:00 AM	74,462	8,625	7,716	7,116	**97,919**
8:15 AM	70,901	8,287	7,519	6,615	**93,322**
8:30 AM	63,533	7,488	6,656	6,095	**83,772**
8:45 AM	55,041	6,689	6,133	5,285	**73,148**
TOTAL	**849,449**	**100,150**	**90,391**	**80,062**	**1,120,052**

If this model is to be useful in the future effort will be needed to rework the network so that it can accept demand from the new ABM. For now, the model is running in a hybrid state using the demand resulting from the Dynamic-ODME with all freeways being modeled microscopically and arterials in mesoscopic given that the arterials would need traffic signals to run accurately in microscopic. The update interval of vehicles is complex in order to better capture driver behavior and reduce the computational burden. In general, every vehicle's position is updated every 0.1 seconds in the microscopic region and 1 second in the mesoscopic.

9.2.4 Computational Requirements

Running computational demanding models and pushing simulation software to its limits has been the norm at the MTO for some time now. Experience has shown that knowing how much computing power you need is essential for large simulations. Run times of simulations can change by orders of magnitude simply because memory was not readily available or the modeler requested too much output to be produced. The research team was able to only test performance on the not-complete TransModeler model we have available. This model did not consume resources to the level of becoming an issue. To understand the requirements, we asked the developers about their experience. According to the developers, even one of their biggest models[28] doesn't use more than 16GB of RAM. Caliper also reported that the CPU tends to be the most contributing factor in influencing the runtimes of TransModeler. It is recommended to use processors with base clock speeds above 3.00Ghz and to seek up to 16 physical cores as Caliper has seen noticeable diminishing returns beyond 8 physical cores and notable beyond 16. The decrease in efficiency beyond 16 cores is notable enough not to be worth the investment and specifically is avoided since computers with more than 16 physical cores per motherboard generally have lower clock speeds and thus increase run times. In addition, while intel's hyper threading can get more logical threads out of each physical core the developers have not seen significant benefits from it and would instead opt for physical cores when available. To put this in layman's terms, TransModeler does not benefit from a large $20,000 server. A regular workstation computer is more than sufficient for nearly every network developed in TransModeler and Caliper is more than willing to share the specs of their workstations which are mostly off the shelf Dell Systems.

9.2.5 Results/Outputs

The following section outlines the steps needed to perform a DTA simulation and the results produced by the software. The steps in which TransModeler performs a DTA are slightly different from those of other simulation suites and thus care must be taken to select the right inputs/outputs between runs.

To start we begin with OD matrices. As is common practice the use of an ODME tool is to calibrate the demand coming in and fix any mistakes it may have. Calibrating the demand entering a network to match location based volume measurements is a debatable practice that unfortunately has become very common. In fact, it has come to be that this form of "calibration" accounts for the majority of the model calibration that takes place. It is conceivable that if appropriate and uniform volume measurement coverage exists, the individual section characteristics like free flow speed and geometric design, as well as global driver behavior characteristics are known and correct, fixing the input demand may be the calibration needed. In the case of this test model developed with the help of Caliper, since the majority of arterials is not included, the volumes on freeway segments results in a reasonably uniform coverage. An ODME tool, whether static or dynamic, should be used with caution and with the assistance of the developers when first being used. Very few ODME procedures are the same and none run in a reasonable amount of time. The main reason Caliper has not released the Dynamic ODME tool is due to the 2-3 month run times on extremely large networks one wrong initialization step can result in a bad

[28] The model referenced was outlined in Task 1 of this project - Jacksonville, FL

result. For the purposes of this model the OD's provided by Caliper are accurate enough until additional geometry is added.

Next, we move into running a Dynamic Traffic Assignment using a Dynamic User Equilibrium approach. In TransModeler the purpose of using a DTA via DUE is to produce "Historical Travel Time" (Table 4) and "Turning Delay" (Table 5) tables instead of outputting result directly. These outputs can then be used as input historical travel time and turning movement tables for future simulation runs. A path file can also be output by the DUE-DTA that is in a proprietary format but contains a record of each path and the links associated with it. A separate file called a "Path Flow Table" as seen in Table 6 is made for each vehicle type and keeps a record of how many vehicles left in each interval using the paths between every OD pair. This path file can be used as an input to the demand for a Stochastic-DTA but this method is not recommended by the developers.

Table 4: TransModeler "Historical Travel Times" Table Example from DUE-DTA simulation run

ID	***AB_ Time_ 0600***	***BA_ Time_ 0600***	***AB_ Time_ 0615***	***BA_ Time_ 0615***	**...**	***BA_ Time_ 0845***	***AB_ Time***	***BA_ Time***
15123	5.02	3.64	4.92	3.57	...	19.67	7.63	3.83
14960	19.73	19.85	19.70	19.92	...	19.78	25.09	19.85
16996	3.09	3.10	3.09	3.10	...	3.12	3.10	3.10
16997	42.78	42.62	42.78	42.69	...	42.78	42.76	42.77
15981	2.32	2.54	2.34	2.50	...	2.53	2.43	2.53
15795	2.14	1.12	2.22	1.33	...	1.31	1.87	1.23
15703	1.70	1.74	1.70	1.73	...	1.75	1.72	1.73
15445	5.60	4.77	5.51	4.82	...	4.89	5.29	4.80
15167	10.26	--	10.08	--	...	--	10.19	--
15170	30.64	--	30.42	--	...	--	51.08	--

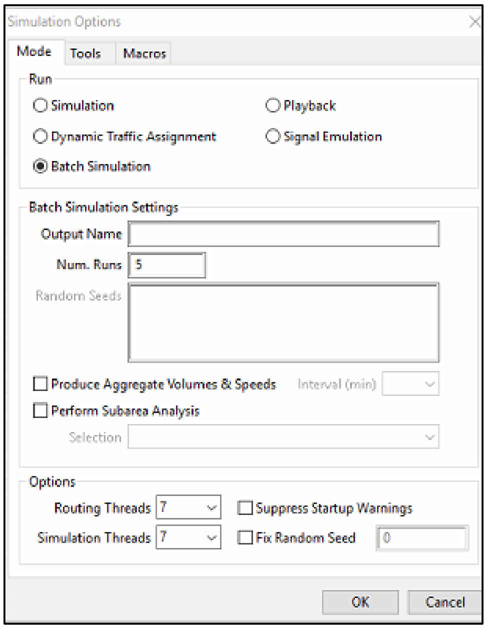
Table 5:TransModeler "Turning Delays" Table Example from DUE-DTA simulation run

From Link	*Node*	*To Link*	*Dir*	*Type*	*Round-about*	*Delay_ 0600*	*Delay_ 0615*	*…*	*Delay_ 0845*	*Delay*
5868	8143	5870	SW	R	--	0.35	0.32	…	1.65	0.73
5868	8143	5869	SW	T	--	0.13	0.23	…	0.55	4.45
5949	15152	5950	S	R	--	--	--	…	0.00	0.01
5949	15152	5919	S	L	--	1.90	1.31	…	1.84	1.76
5950	15152	5949	SE	L	--	--	0.00	…	0.03	0.09
5950	15152	5919	SE	T	--	0.65	0.60	…	0.91	1.01
5919	15152	5949	NW	R	--	1.14	0.50	…	0.30	0.60
5919	15152	5950	NW	T	--	0.03	0.28	…	0.55	0.58
5949	8594	5952	N	T	--	0.80	0.22	…	0.39	1.81
5949	8594	5920	N	L	--	2.03	0.03	…	0.32	0.30

Table 6:TransModeler "Path File" Table Example from DUE-DTA simulation run

OriType	*OriID*	*DesType*	*DesID*	*OriLink*	*Path*	*Interval 1*	*Interval 2*	*…*	*Interval 12*
Centroid	7939	Centroid	32188	-11325	1058375	1	0	…	0
Centroid	2097	Centroid	1993	-8809	1056972	1	0	…	0
Centroid	6401	Centroid	6136	1357	1058075	1	0	…	0
Centroid	1974	Centroid	10418	9014	1056342	1	0	…	0
Centroid	2003	Centroid	1801	-8306	4426737	1	0	…	0
Centroid	2054	Centroid	2057	7460	1056741	1	0	…	0
Centroid	1942	Centroid	1949	7986	1056123	1	0	…	0

Those three items are all that is produced from the DUE-DTA. To obtain statistics such as speed and volume a second simulation needs to be run using the described outputs of the DUE-DTA. This simulation is often referred to as a stochastic-DTA, in the case of TransModeler’s GUI this would be either a “Simulation” or “Batch Simulation”. “Simulation” would run a single DTA assignment while “Batch Simulation” would multiple simulations, independently with the option to have some outputs aggregated automatically. Since this project focuses on DTA the team feels it is important to note the discrepancy in vocabulary used by TransModeler and the research team. As seen in Figure 13 the TransModeler GUI explicitly calls out Dynamic Traffic Assignment as a simulation method. This contradicts the research team’s terminology since both the simulation and batch simulations are performing DTA in their route choice steps. To clarify this the team would refer to the “Simulation” run as a Stochastic DTA, the “Dynamic Traffic Assignment” as Dynamic User Equilibrium with DTA, “Batch Simulation” as Batch Stochastic DTA, assuming that the model is using OD matrices as its input, which is required for DTA.


Simulation Options

Mode Tools Macros

Run

○ Simulation ○ Playback

○ Dynamic Traffic Assignment ○ Signal Emulation

● Batch Simulation

Batch Simulation Settings

Output Name

Num. Runs 5

Random Seeds

□ Produce Aggregate Volumes & Speeds Interval (min)

□ Perform Subarea Analysis

Selection

Options

Routing Threads 7 □ Suppress Startup Warnings

Simulation Threads 7 □ Fix Random Seed 0

Figure 13: TransModeler Simulation Options GUI

While the MTO research team has experiences with multiple simulation model suites TransModeler has proven difficult to get information out of. Generally, outputs are stored into proprietary ".bin" files if there are any at all since you need to be extremely vigilant from the start as to what outputs you would like since much of the information is never written to permanent storage and impossible to get after the simulation is complete. While TransModeler has a numerous number of reporting tools they would take quite a bit of effort to learn how to use them. The MTO generally prefers simulation output be sent directly to an SQL database so that information can be queried directly using the modelers preferred program. The ".bin" files can be opened by TransModeler and saved to numerous formats (Excel, CSV, SQL) but this can be labor intensive.

The results shown in Figure 14 were extracted with the following inputs to 5 independent stochastic DTA runs:

- OD Matrices from the Dynamic ODME
- "Historical Travel Time" and "Turning Delay" tables from a DUE DTA that ran for 25 iterations and reached a Rgap value of 0.05
  - Run Time = 13.25 hours
    - 8.5 minutes can be saved per iteration where the R-gap is not calculated
  - 53061 trips were still en-route when the simulation ended.
  - 1084973 trips were successfully completed.

The rules show that for a model that has only had minimal effort put into it replicates reality to some degree. While considering some of the more extreme mismatched cases these generally turn out to be coding errors such as missing turning movements or lane coding errors. To produce that graph one must first join output from the poorly named file “Historical Travel Times”, which is different and not to be confused with the one output by the DUE-DTA, to a segments data view. Once joined it can be saved into any useful format (excel, “.bin”, CSV, etc.) and will contain segment speeds and volumes from each interval (in this case 15 minute) specified for that simulation.

1. Batch Simulation 1
    a. Run Time = 40.5 minutes
    b. 75756 trips were still en route when the simulation ended.
    c. 1074836 trips were successfully completed.
2. Batch Simulation 2
    a. Run Time = 40.81 minutes
    b. 75344trips were still en route when the simulation ended.
    c. 1074919 trips were successfully completed.
3. Batch Simulation 3
    a. Run Time = 39.96 minutes
    b. 75319 trips were still en route when the simulation ended.
    c. 1074456 trips were successfully completed.
4. Batch Simulation 4
    a. Run Time = 40.23 minutes
    b. 77137 trips were still en route when the simulation ended.
    c. 1072927 trips were successfully completed.
5. Batch Simulation 5
    a. Run Time = 41.46 minutes
    b. 73630 trips were still en route when the simulation ended.
    c. 1076973 trips were successfully completed.

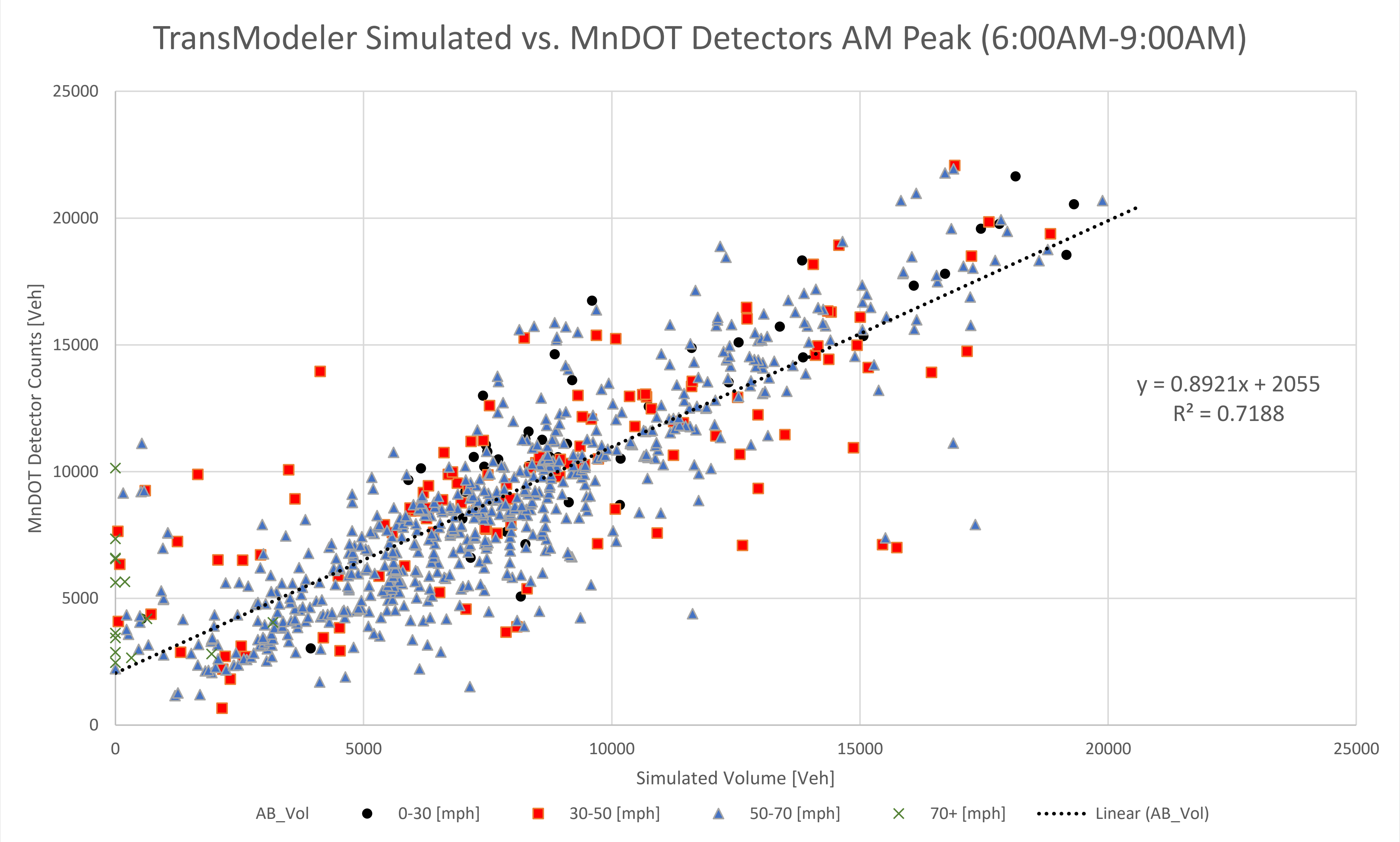


Figure 14: TransModeler Simulated Volumes vs. MnDOT Detector Counts during AM Peak (6:00AM-9:00AM) Colored by Simulated Speed

9.3 PROPOSED WORK PLAN AND SCHEDULE FOR COMPLETING THE TRANSMODELER MODEL

Based on the findings of this report and the expertise of the research staff it would be beneficial to MnDOT and other entities to pursue the creation of a calibrated Twin Cities DTA model in TransModeler. Caliper and the research team have already created a rough network with limited resources that is showing great potential. By building the model in TransModeler there is also room for expansion and additional detail that cannot be done in the current DynusT DTA model without moving the network to a different software package.

Specific items would need to be addressed for this model to be successful. The first would be to update the geometry to match the level of coverage seen in the Activity Based Model (ABM) since it will most likely be providing the demand for the region. Effort was given to get pricing information from AirSage on the cost to acquire OD's on the same level as the ABM model. Unfortunately, they were not able to provide an exact price for OD data that spans the scope of the ABM due to the complex pricing structure. However, in a recent study of Highway 10 in Anoka AirSage provided OD data for a simulation model. This model had 30 TAZ's and provided one days' worth of data broken down by hour for $10K. Since the ABM has 3300+ TAZ's and more than a days' worth would be needed it could be theorized that the price would increase substantially. While this is a feasible path for obtaining OD data the ABM would provide "free" data as it is already being established as a tool for the region.

The second task would be the collection of arterial data for a better ODME result and to assist in calibration. Methods and potential sources of this information were outlined in Section 9. The third would be to enter traffic control information to simulate delay at intersections and ramps. This task can be approached in two different ways. Either signal timings can be generated by TransModeler using tools built in the next stable release or current signal timings could be obtained from the various entities in the region. The latter would be influenced by the implementation of a new central signal system that is being implemented in the Fall of 2016 that may greatly reduce this normally time intensive effort. The final task would be to calibrate and validate the model on a regional scale.

All in all, Caliper has put forth around 160 man hours towards the technical aspects of the model such as network geometry and scenario preparation. They also had several months' worth of computing effort while running the Dynamic Origin Destination Matrix Estimation tool to better match the freeway network counts. They noted that if this procedure were to be implemented again for the whole regional network, including arterials, they would need additional network counts and anywhere from 3-6 months for the Dynamic ODME tool to reach a stable convergence.

Based on our conversations with the development team and their past experiences with building Mesoscopic and Microscopic models of this size they are confident a model could be achieved within 1 to 1.5 years. They are also confident, if requested, that TransModeler can model everything within the 494/694 ring road in microsimulation as they have already done in a model of Phoenix, Arizona for the

Maricopa Association of Governments. In that model, they developed a 500mi^2 microsimulation model with a .1 second time step and more than 1800 signals. The time line would of course depend on the level of detail requested and the amount of calibration warranted. For example, if the model was designed to reflect current conditions with the freeways in Microscopic, arterials in Mesoscopic, the ability to import demand data from the ABM, and a framework for updating arteries (signals, additional geometric detail, etc.) Caliper would estimate around 3700 man-hours at approximately $650K to fully calibrate it. If calibration and validation did not have to be as refined, then their estimate could be as much as half in terms of both hours and cost. This cost of course is only a “ballpark” estimate, the price could vary greatly based on the needs of the client, and the amount of analysis required as the above figure only includes the model creation with no analysis besides validation.

CHAPTER 10: CONCLUSIONS

As was made clear throughout this report, there is a need for a simulation model of the Twin Cities metro area, or large portion of it, that can replicate the dynamic nature of traffic. Long gone are the days where increasing the number of lanes or adding new freeways was the solution to the never-ending congestion problem. This capacity increase approach was generally captured well within travel demand models. Now with systems such as open access MnPASS lanes it is becoming increasingly debatable, and in the research team's opinion impossible, to model these complex traffic patterns using static travel demand models. Thus, it is necessary to create a model that can capture these issues but can still be retooled for future projects much like our Metropolitan Council's Travel Demand Models and that is where Meso-DTA or even Micro-DTA models fill the void.

During the beginning of this project, and due to the need for MnDOT to have a Meso-DTA model to plan the 2017 construction year, the development of such a model could not wait for the findings described in this report. Regardless, research into this issue was still needed and thus the team first began with finding local (USA) examples of large Meso/Micro-DTA models, followed by gathering local stakeholder opinions, as well as researching commercially available software through their identified relevant features and compared the pros/cons of singular models to project sized needs. At this point, the project was originally going to create a framework/guideline for the creation of a meso-DTA model but given that a Meso-DTA model was already in development, the TAP was given two choices. Either the team could investigate the Meso-DTA model produced for MnDOT and identify its strengths and weaknesses, or it could pursue an opportunity that presented itself midway through the project where the developers of TransModeler (Caliper) would help the team produce a proof of concept model. After careful consideration, the TAP decided for the latter and thus a concept model was pursued and produced as the final task of this project.

In the first task of this project, the team investigated several DTA models around the United States. It was found that there are numerous successful DTA models in use and four representing a different software and each were chosen for a detailed examination; the Manhattan Traffic Model modeled in Aimsun, the San Francisco DTA Model modeled in Dynameq, The Detroit DTA Model modeled in DynustT, and the Jacksonville DTA model modeled in TransModeler. The common theme among these models was that data collection is extremely important. Mesoscopic and Microscopic models need extensively more data due to their detailed account of traffic patterns. These data include items such as signal timing information, freeway traffic counts, arterial traffic counts, and average speed data. In the Twin Cities, freeway data are already available, but items such as arterial traffic counts would need to be purchased from external vendors or an institutional change needs to be made to collect data on arterials in the same way as MnDOT does for the freeways.

While the stakeholders had many opinions, which were outlined earlier, it was imperative that for a model to be useful it must be able to run in a reasonable amount of time. Many have avoided building larger models due to the increased manpower needed to create them and the extensive computing power/time needed to run them. While some of these such as computing power are increasing with time due to technological advancements the number of hours in a day are not. Therefore, for a model to be useful and for traffic studies to remain on schedule, the model needs to be able to run in under a day to be useful since calibration still comprises a lot of “Guess and Check” type approaches. The second biggest issue with Mesoscopic and Microscopic simulation for many stakeholders was how to gather information for arterial traffic signals since the represent a significant portion of delay. While the different software suites could consider delay based on the signal timings, they are hard to obtain on a regional level and can require extensive manual effort to add them into a model. Every stakeholder expressed the desire to have one universal format for the signal timing data so that every signal could be programmatically coded into the model and thus give the most accurate account of delay at those intersections.

One of the longest tasks of this report was creating a comprehensive list of the features from each of the commercially available software suites and identifying their strengths and weaknesses. The team realizes that this list is only a snapshot in time, these software packages are always evolving, and some of these features may have changed since they were looked at. However, this list is still a useful tool since it can be used to narrow down the software that could fit a project’s specific needs and while the no features may change over time the yes features will most likely not, and thus if it has a specific feature now, it will most likely have that feature in the future. The biggest takeaway from this task is that two of the software suites, Aimsun and TransModeler, stood out as being both feature rich and able to have small portions extracted from the model that would allow local consultants to expedite a model’s development for a localized issue without having to run an entire regional model.

Over the course of this project, the research team considered whether a Meso-DTA model should take the form of multiple smaller models that are linked together to eventually create a regional model or if a single regional model should be created from the beginning. Stakeholders weighed in on this possibility, and while most would like to maintain the ability to use the software package of their choosing, they would entertain the idea of using a different suite provided it included a Meso-DTA model. They noted that the Metropolitan Council’s Travel Demand Models do not necessarily have to run in CUBE since all the legwork is done outside of it, but it was cheaper for the consultants to use CUBE instead of attempting to transfer it into a new program. Therefore, given the willingness of stakeholders to use a different software if a tool was readily available and the difficulty of mixing software packages, the team concluded that the most feasible approach to developing a regional meso-DTA model would be to create it in a single software packages as one all-encompassing model.

As noted earlier, two simulation applications were identified as having all the required features at the required accuracy and realism to cover the needs of a MESO-DTA user, Aimsun and TransModeler. In the case of AIMSUN, the research team has extensive experience and has already developed a Twin Cities-wide hybrid DTA model as part of an earlier project. During that research, it was concluded that the current execution speed of such a model was too slow for wide use and adoption by MnDOT, although recent improvements in the simulation software and computational power have shown a decrease in the amount of time needed for the model to converge. During this project, the relatively new TransModeler software was investigated and shown to possess impressive performance in terms of speed. In addition, Caliper, the software developer, offered to assist the research team in testing some of the software features under network size realistic conditions. The firm's help was offered specifically in building a large network geometry, importing a sample of MnDOT's arterial traffic control information, and most importantly, importing and adjusting the network demand information as received from the Metropolitan Council's Regional Travel Demand Forecasting Model. While the team would have preferred the demand come from the new Activity Based Regional Planning Model, the integration procedure would have been considerably more complex due to the increased level of detail and would have hindered the project's timeline. Nevertheless, the older RTFDM provided a seed matrix that the Dynamic-ODME procedure was able to use to calculate a more refined demand, based on the freeway detectors.

During this proof-of-concept, reasonable benchmarks in terms of effort needed were reported. The Run Times of the model are also reasonable although very close to the limit specified by the stakeholders interviewed. Specifically, in contrast to other applications, TransModeler runs the DUE DTA simulation independently from other parts of the process, and it is especially notable that this process does not produce results that would directly allow validation of the results. The validation can only be achieved if the "equilibrated" travel times produced by the DUE DTA are imported into a Stochastic DTA simulation. As stated by the developers, one iteration of the Stochastic DTA cannot be considered a repetition of the last DUE DTA iteration because the DTA algorithm involved in the two approaches is not the same but would more or less be similar. The proof-of-concept network included a very detailed network of about half of the city of Minneapolis and one-third of St. Paul, as well as all the freeways and major highways in the metro area. For this network, the DUE DTA of the three-hour morning peak period (6a.m. to 9a.m.) takes approximately 13 hours to complete, while each iteration of a Stochastic DTA of the same period takes approximately 40 minutes. These times are very reasonable for the network size and detail. The result from this not accurately calibrated model allows us to predict that successful full network is feasible. However, the process of producing these results and others from TransModeler was a very involved process that needs to be made simpler for the model to be successful.

REFERENCES

(1) Chiu, Yi-Chang, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks, *A Primer for Dynamic Traffic Assignment*, (National Academies of Sciences, Engineering, and Medicine; Transportation Research Board; Technical Activities Division, 2010) DOI: https://doi.org/10.17226/22872

(2) Transport Simulation Systems. “Aimsun 8 Dynamic Simulators Users’ Manual” (Software Manual Transport Simulation Systems, 2014).

(3) B. Ran, and D. Boyce, (1996), Modeling Dynamic Transportation Networks, Springer-Verlag.

(4) B. N. Janson, (1991), Dynamic Assignment for Urban Road Networks, *Transpn. Res. B*, Vol. 25, Nos. 2/3, pp. 143-161

(5) Cambridge Systematics. “Manhattan Traffic Model Specification Report” (Final Technical Memorandum, June 2010). PDF.

(6) Cambridge Systematics. “Manhattan Traffic Model Network Development” (Final Technical Memorandum, June 2010). PDF.

(7) Parsons Brinckerhoff & San Francisco County Transportation Authority. “Final Calibration & Validation Report” (Technical Memorandum*,* Nov 2012). PDF.

(8) Parsons Brinckerhoff & San Francisco County Transportation Authority. “Final Methodology Report” (Technical Memorandum*,* Nov 2012). PDF.

(9) Parsons Brinckerhoff. “I-96 Dynamic Traffic Assignment (DTA) Model”*.* (Technical Memorandum, July 2012). PDF.

(10) S. E. Binkowski, J. E. Hicks. *Development and Use of a 700-square mile DTA Model for Corridor Maintenance of Traffic Decision-Making.* (Transportation Research Board, Annual Meeting, Paper #14-3816, 2014) http://docs.trb.org/prp/14-3816.pdf

(11) Caliper Corporation. “TransModeler Dynamic Traffic Assignment Model for NERPM ABM.” (North Florida TPO, Technical Memorandum, 2013). PDF.

(12) J. Sloboden, J. Lewis, V. Alexiadis, Yi-Chang Chiu and Eric Nava *Traffic Analysis Toolbox Volume XIV: Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling*, 2012. PDF. Publication #: FHWA-HOP-13-015

APPENDIX A – CASE EXAMPLES

This appendix outlines the four specific models that represent the state-of-the-art in large scale mesoscopic simulations. It also covers the capabilities of several of the software suites analyzed as part of this report. Many of these details were obtained directly through the software vendor and/or the clients involved and thus their reports are not publicized.

Aimsun – New York Manhattan Traffic Model

All information used in this section was derived from unpublished technical memorandums (5, 6).

The New York Manhattan Traffic Model (MTM) is a large Meso-Micro DTA model developed for the New York City Department of Transportation (NYCDOT) by Cambridge Systematics. The MTM was designed to work with the New York Metropolitan Transportation Council's Best Practice Model (BPM), which is a large regional macroscopic multimodal 4-step travel demand model. The model is being concurrently improved and maintained so that it can provide insight into future projects in the region.

The MTM was developed because NYCDOT needed a model that could accurately replicate complex real world properties such as: managed lane strategies, double parked cars, pedestrian movements, taxi movements, queue propagation, bus stops, on street parking, and signal coordination. The first study to use the MTM upon its completion was a traffic signal priority project that studied the improvements of signal priority on buses along 34th St.

Geometry

The model was developed in various stages of detail to cover the greatest number of projects. The MTM itself was originally designed around four main study areas: the core, the microsimulation area, and the primary and secondary study areas. The core area covering 3.2 square miles consists of all streets bounded by 44th street to the north, 28th street to the south, the Hudson River to the west and the East River to the east. The core study area was programmed to the level of detail necessary for microscopic or mesoscopic simulation. This included characteristics such as signal timings, roadway geometry, and curbside activity. The microsimulation area is a 1.1 square mile subset of the core area that was explicitly developed to be run at the microsimulation level. This region is bounded by 37th and 32nd street to the north and south along with the Hudson and East Rivers to the west and east, respectively. The primary study area covers another 6.5 square miles not covered by the core area, bounded by 66th and 14th street to the north and south as well as the two rivers described previously to the east and west. The purpose of the primary study area was to capture the local travel patterns in and out of the core area and to capture the traffic operational changes from no-build projects. Again all streets are represented in the study area but in this case "coarser" methods were used to describe the operational details of those section. The secondary area covers another 23.5 square miles on Manhattan Island that were not covered in the other study areas. It is bounded by 179th street to Battery Park in the north and south, while still bounded by the Hudson and East Rivers. It also covers select areas of the boroughs as well as New Jersey. This area was designed to capture the regional travel patterns going into and out of the primary area as well as also capturing operational changes in no-build scenarios. The whole extent of the network can be seen in Figure.

All of the geometrics mentioned above were implemented as follows. First, the BPM and NYC LION[29] GIS files were merged to provide a base network for the MTM. These networks were designed and implemented based on a macroscopic modeling process and therefore almost all links were unsuitable for the lower level microscopic simulator without modification. For example, while all avenues were included in the BTM model, many were collapsed into “composite streets”. These streets were designed to replicate several streets in one link which is acceptable from a macroscopic modeling perspective but not in a microscopic network. All of these issues were resolved by importing aerial imagery, field surveys, and local knowledge to enhance the network to the level of detail required. This process was completed for the entire primary study area (which includes the core and microsimulation area) while the secondary area was left without modification from the BPM model because it was sufficient for the mesoscopic simulator.

[29] GIS data: A single line street base map representing New York City's streets and other linear geographic features, along with feature names and address ranges for each addressable street segment. This dataset also includes a nodes file for representing intersection at line junctions. LION is a name for the dataset and not an acronym.

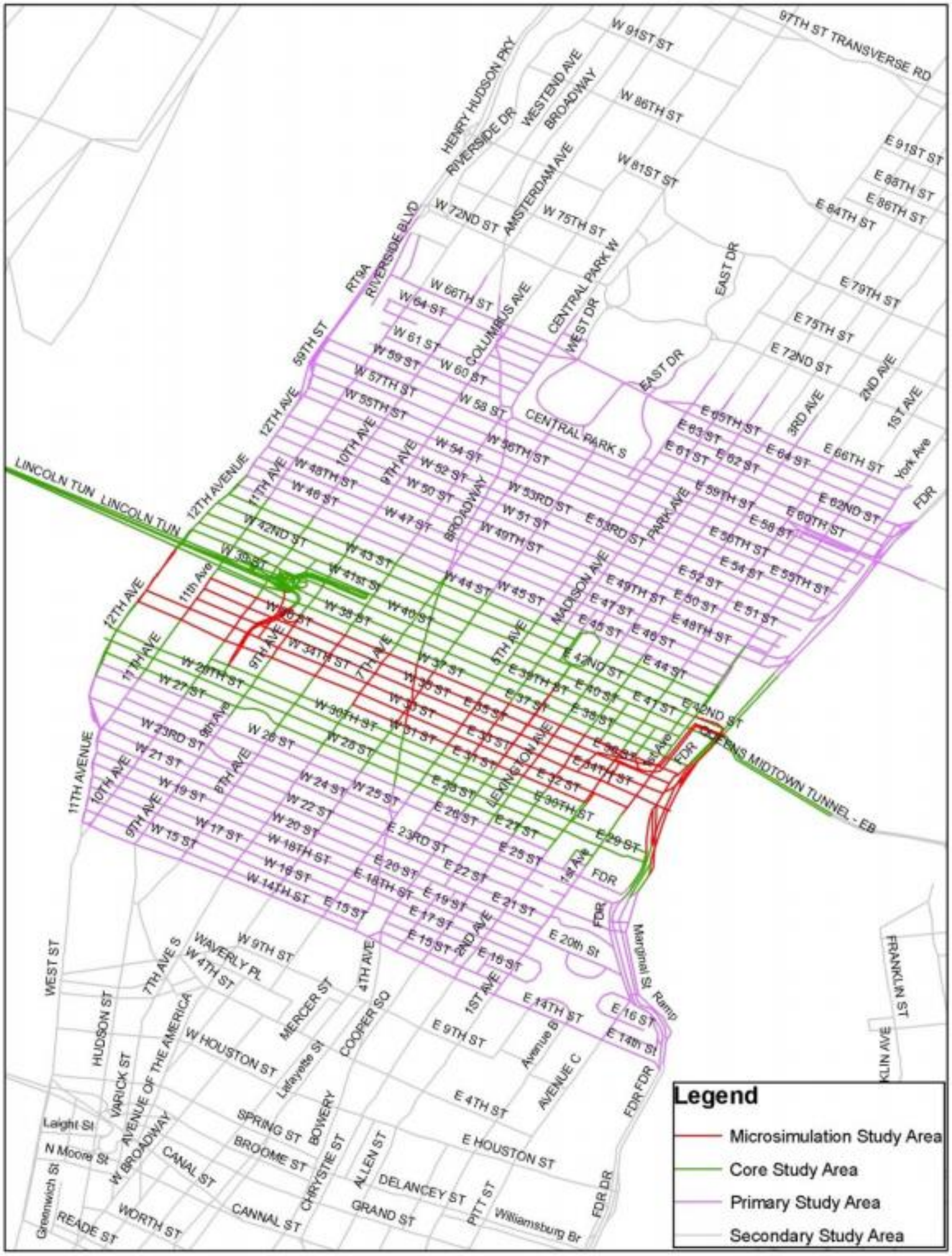


Figure 15: MTM model colored by study area

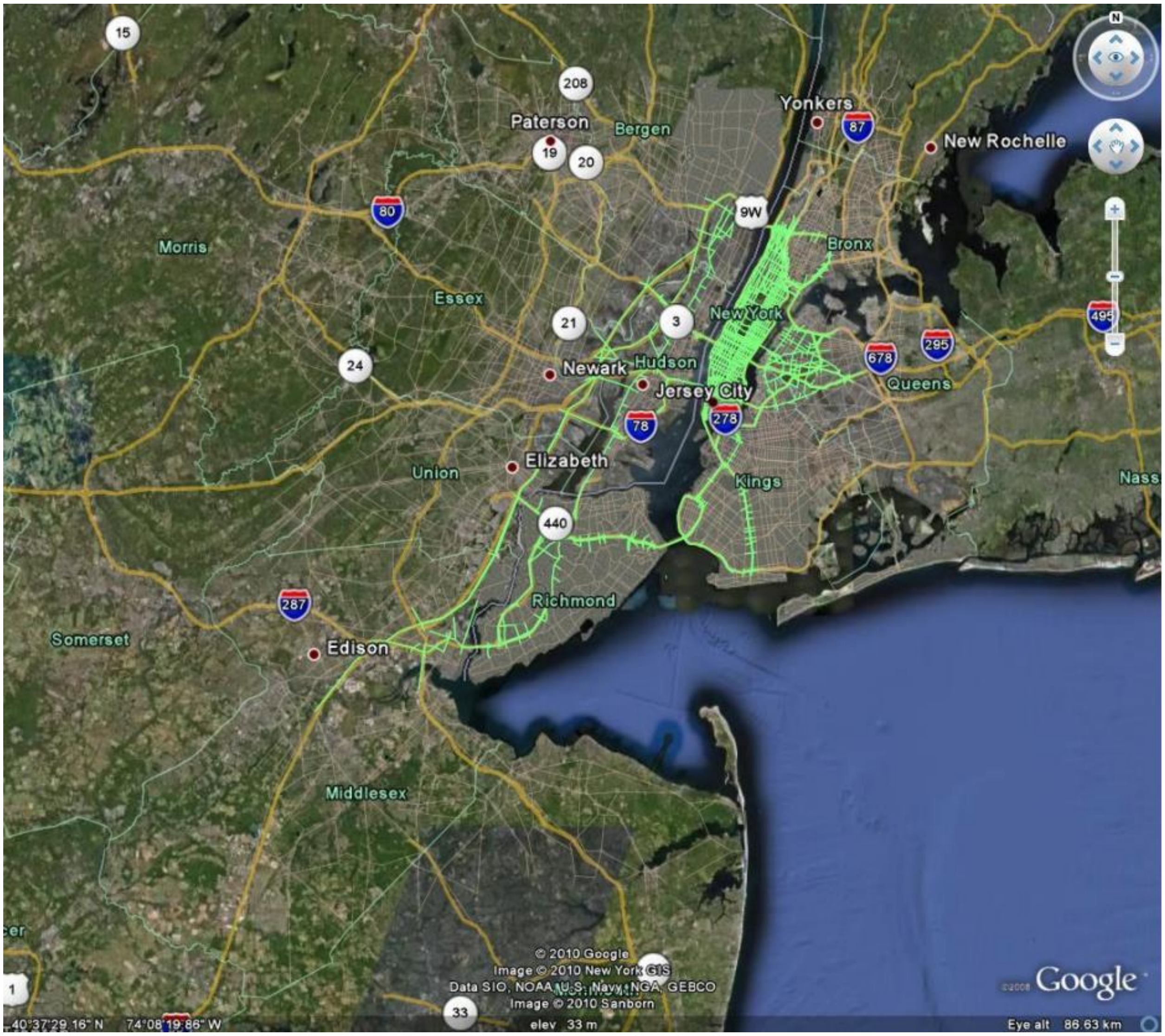


Figure 16: Full MTM network on top of google maps

Control

As was seen in many implementations of DTA networks, the need to input traffic control information was essential to modeling intersections in downtown Manhattan. The model developers were faced with a common issue that plagues many micro, meso, and DTA simulations: no centralized database of signal control plan data exists for the region. This prevents anyone from getting signal information easily and virtually guarantees that there will not be a consistent format between data received from varying municipality, county, and state agencies. Of the plans received from NYCDOT, each had to be converted into three text files that could then be imported into AIMSUN using a script in order to avoid implementing them manually through the graphical user interface (GUI). If plans were not provided for intersections with control but were within a previous model, such as the Green-Light model and the Third Water Tunnel model, those plans were extracted and imported.

For intersections that did not meet the above criteria a third method was developed by using a pattern that arose from the signal control plans. In total, nine distinct patterns were derived and are listed in Table . For all of these combinations it was assumed that the signals were simple two-phased with no exclusive left or right turns and 90 second cycle lengths.

Table 7: MTM-DTA models cross street combinations with signal timings

		EAST/WEST TWO WAY		**ONE WAY EASTBOUND**		**ONE WAY WESTBOUND**	
			G/Y/R		*G/Y/R*		*G/Y/R*
NORTH/SOUTH TWO WAY	*N/S*	1	43/3/2	2	45/3/2	3	45/3/2
	E/W		37/3/2		35/3/2		35/3/2
ONE WAY NORTHBOUND	*N/S*	4	40/3/2	5	49/3/2	6	49/3/2
	E/W		40/3/2		31/3/2		31/3/2
ONE WAY SOUTHBOUND	*N/S*	7	40/3/2	8	49/3/2	9	49/3/2
	E/W		40/3/2		31/3/2		31/3/2

Finally, if a signal could not be retrieved/reproduced using any of the previously mentioned methods then the signal control plans were created to accommodate the turning movement demands estimated during the recreation of the origin/destination files.

Demand

As discussed in the geometrics section, the MTM was first built from the BPM. This also allowed the MTM to be linked to the numerous databases that the BPM relies on to do its traffic forecasting. By linking the MTM to those databases it is able to use the BPMs Origin Destination Matrix Estimation (ODME) to generate the demand needed for the model to run. However due to the additional refinement of the network geometry in the primary study area the original Traffic Analysis Zones (TAZ) were subdivided into block level zones. Those zones were not only refined in their size but in how they connected to the network. All of the centroid connectors that distribute demand from the TAZ onto the roadway were manually reconnected at midblock locations instead of at the intersections to avoid vehicles disappearing and appearing at those intersections. Those connectors also limited which vehicles could enter onto a particular class of street. This was largely to prevent passenger vehicles from ending their trips on avenues that generally do not have parking.

Calibration/Validation

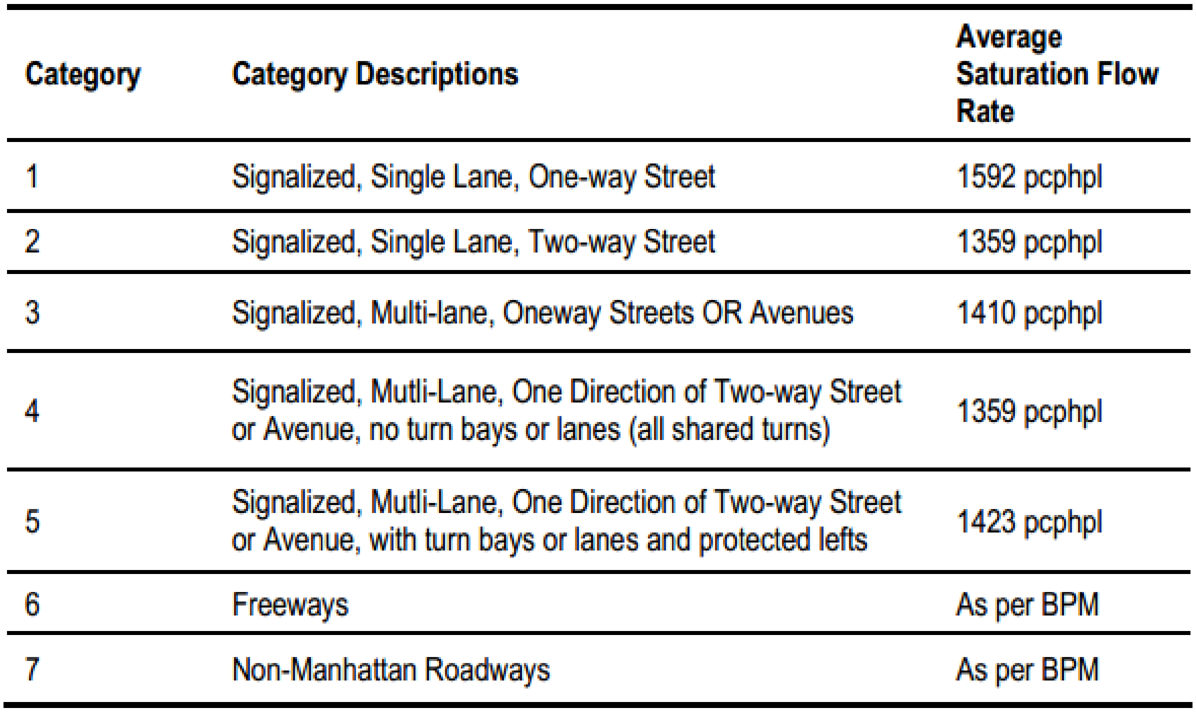
In order to calibrate and validate a model the size of the MTM significant field data is necessary. The Field data that was collected and associated to the MTM network included segment geometry, curbside activity, parking regulations, and turning geometry observations. The MTM was also associated to databases containing data such as Automatic Traffic Recorder counts, Turning Movement Counts, and travel time and aerial queue observations. However, due to the level of effort required to gather all of this information core areas were prioritized. For the core region all of the above information was collected for every intersection and roadway segment by field personnel for all time periods (AM, MD, PM). Throughout the rest of the region aerial photos were used to construct geometry, and parking regulations were not collected. In addition curbside activities were only collected along the avenues and not to the same level of detail as for the core region.

In addition to collecting these data, tests were performed on small test networks to adjust certain parameters, such as Reaction Time at Stop, Reaction Time at Traffic Light, pedestrian flow, taxi percentages, and percentage of left turns. Specifically tests were done to adjust the saturation flow rates seen in the test models to those seen in the real world based off a previous capacity report[30] of the Manhattan area. The results of these test networks are seen in

Table except for category 6 and 7 whose saturation flow rates are taken directly from the BBM.

[30] The Route 9A Reconstruction Project, The Analysis of Capacity Values in Manhattan with Emphasis on the West Side and the Route 9A Corridor, Urbitran Associates, July 1990

Table 8: MTM model saturation flow rates by category



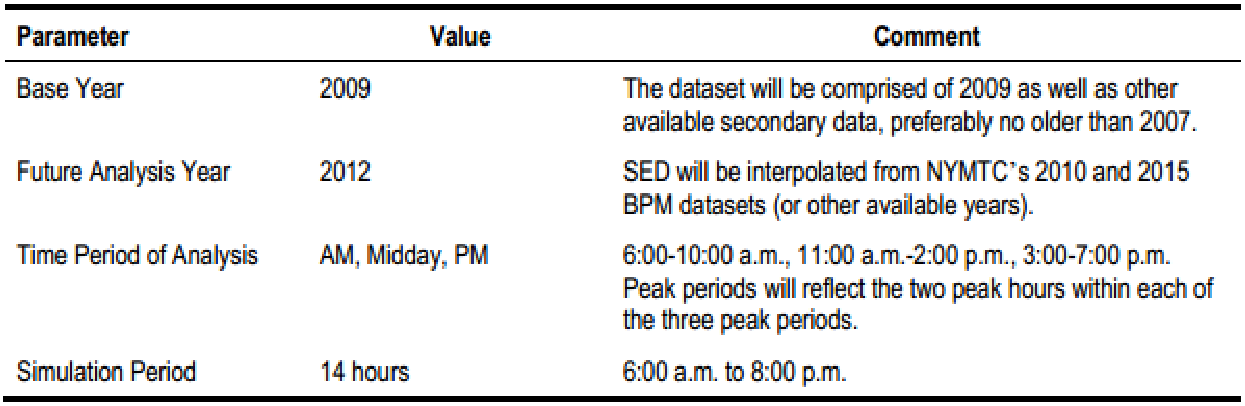
Category	Category Descriptions	Average Saturation Flow Rate
1	Signalized, Single Lane, One-way Street	1592 pcphpl
2	Signalized, Single Lane, Two-way Street	1359 pcphpl
3	Signalized, Multi-lane, Oneway Streets OR Avenues	1410 pcphpl
4	Signalized, Mutli-Lane, One Direction of Two-way Street or Avenue, no turn bays or lanes (all shared turns)	1359 pcphpl
5	Signalized, Mutli-Lane, One Direction of Two-way Street or Avenue, with turn bays or lanes and protected lefts	1423 pcphpl
6	Freeways	As per BPM
7	Non-Manhattan Roadways	As per BPM

Since many of the features that were to be modeled were not readily available in Aimsun (parking which blocks turn lanes, double parking, etc.) the programmers used the Application Programming Interface to add these details to the network. For example, when a dedicated right turn lane is blocked by a stopped cab, vehicles that wish to turn right must do so from a through lane. This type of maneuver does not occur all the time so programming the through lane as shared would be inappropriate. They solved it by dynamically creating a shared lane whenever the right lane was used for stopping.

With all of the elements being simulated and saturation flow rates established the validation and calibration could be conducted. A three-step strategy was used for calibrating traffic operations in the MTM model. First a capacity calibration was performed to identify the values for the capacity adjustment parameters to reproduce observed traffic capacities in the field. A global, network–wide calibration is performed first, followed by local, link-specific fine-tuning. The second was a route choice calibration performed with the route choice parameters, where a sample of OD pairs is selected and the validity of the paths is reviewed based on local knowledge. The third was an operational calibration where the overall mesoscopic and microscopic model estimates of system performance (travel times and queues) were compared to field measurements. Fine-tuning adjustments are made to selected variables to enable the model to better match the field measurements. The validation criteria and acceptable calibration levels are summarized in

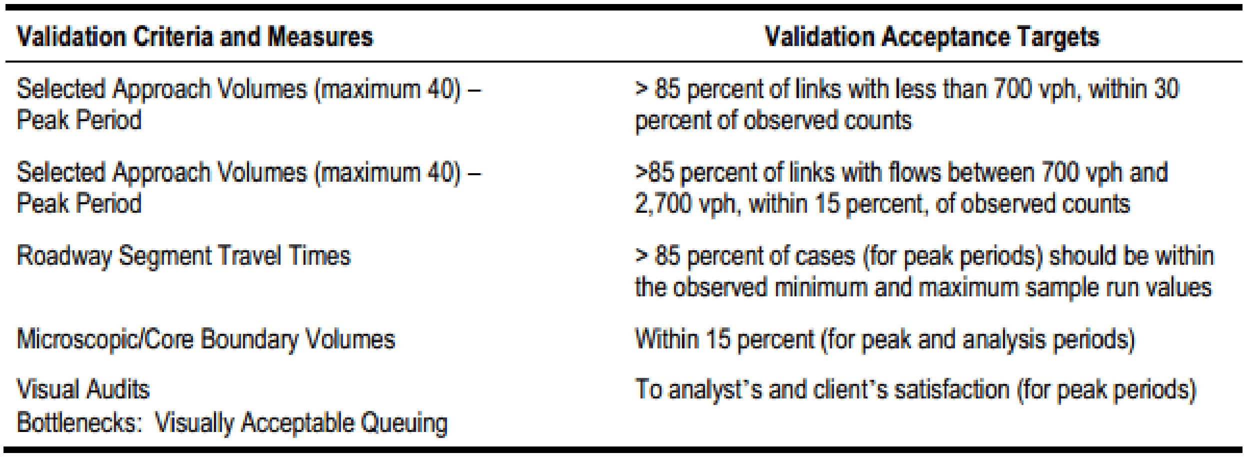
- Table 7 for the 4 defined analysis zones.

Table 9: Summary of analysis settings in MTM model



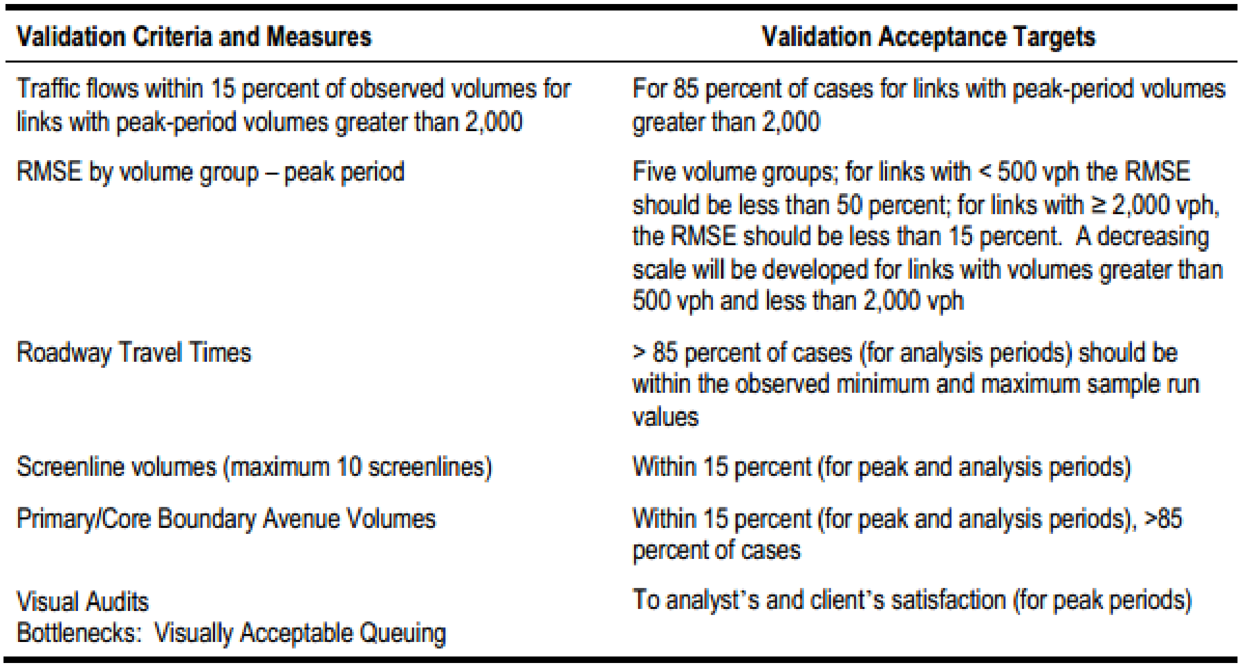
Parameter	Value	Comment
Base Year	2009	The dataset will be comprised of 2009 as well as other available secondary data, preferably no older than 2007.
Future Analysis Year	2012	SED will be interpolated from NYMTC's 2010 and 2015 BPM datasets (or other available years).
Time Period of Analysis	AM, Midday, PM	6:00-10:00 a.m., 11:00 a.m.-2:00 p.m., 3:00-7:00 p.m. Peak periods will reflect the two peak hours within each of the three peak periods.
Simulation Period	14 hours	6:00 a.m. to 8:00 p.m.

Table 10: Summary of microscopic validation criteria and targets for MTM Model



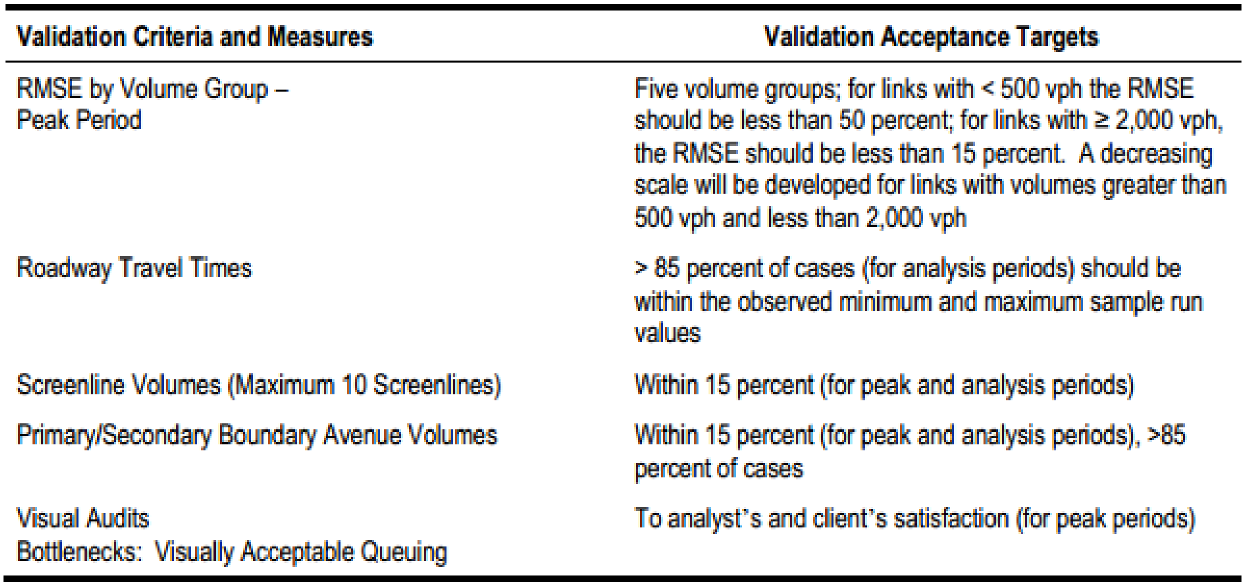
Validation Criteria and Measures	Validation Acceptance Targets
Selected Approach Volumes (maximum 40) – Peak Period	> 85 percent of links with less than 700 vph, within 30 percent of observed counts
Selected Approach Volumes (maximum 40) – Peak Period	>85 percent of links with flows between 700 vph and 2,700 vph, within 15 percent, of observed counts
Roadway Segment Travel Times	> 85 percent of cases (for peak periods) should be within the observed minimum and maximum sample run values
Microscopic/Core Boundary Volumes	Within 15 percent (for peak and analysis periods)
Visual Audits Bottlenecks: Visually Acceptable Queuing	To analyst's and client's satisfaction (for peak periods)

Table 11: Summary of mesoscopic validation criteria for the core area of the MTM



Validation Criteria and Measures	Validation Acceptance Targets
Traffic flows within 15 percent of observed volumes for links with peak-period volumes greater than 2,000	For 85 percent of cases for links with peak-period volumes greater than 2,000
RMSE by volume group – peak period	Five volume groups; for links with < 500 vph the RMSE should be less than 50 percent; for links with ≥ 2,000 vph, the RMSE should be less than 15 percent. A decreasing scale will be developed for links with volumes greater than 500 vph and less than 2,000 vph
Roadway Travel Times	> 85 percent of cases (for analysis periods) should be within the observed minimum and maximum sample run values
Screenline volumes (maximum 10 screenlines)	Within 15 percent (for peak and analysis periods)
Primary/Core Boundary Avenue Volumes	Within 15 percent (for peak and analysis periods), >85 percent of cases
Visual Audits Bottlenecks: Visually Acceptable Queuing	To analyst's and client's satisfaction (for peak periods)

Table 12: Summary of mesoscopic validation criteria for the primary area of the MTM



Validation Criteria and Measures	Validation Acceptance Targets
RMSE by Volume Group – Peak Period	Five volume groups; for links with < 500 vph the RMSE should be less than 50 percent; for links with ≥ 2,000 vph, the RMSE should be less than 15 percent. A decreasing scale will be developed for links with volumes greater than 500 vph and less than 2,000 vph
Roadway Travel Times	> 85 percent of cases (for analysis periods) should be within the observed minimum and maximum sample run values
Screenline Volumes (Maximum 10 Screenlines)	Within 15 percent (for peak and analysis periods)
Primary/Secondary Boundary Avenue Volumes	Within 15 percent (for peak and analysis periods), >85 percent of cases
Visual Audits Bottlenecks: Visually Acceptable Queuing	To analyst's and client's satisfaction (for peak periods)

Table 13: Summary of mesoscopic validation criteria for the secondary study area of the MTM

Validation Criteria and Measures	Validation Acceptance Targets
Sum of Link Flows	Within 5 percent
Selected OD (Maximum 20) Travel Times within 25 Percent	OD modeled travel times versus taxi data travel times (analysis period)
Volumes at Regional Facilities (e.g., QMT, BBT, LT, etc.)	Between 10 percent and 15 percent of observed counts (peak period and analysis period). Observed counts reflect values published in the 2008 NYCDOT Bridge and Tunnels Report
Screenline Volumes (Maximum 15 Screenlines, inclusive of Primary Area)	Within 15 percent (for peak and simulation periods)

Summary

A summary of essential model statistics was provided in

Table to allow a quick comparison between the different models. It should be noted that when this model was developed, Aimsun was on version 6 and, as a result, a step had to be introduced in order to simulate the microscopic and mesoscopic portions together. Now Aimsun is on version 8 which has significant computational improvement as well as the addition of hybrid modeling which, as the name suggests, combines mesoscopic and microscopic in one traffic assignment. This hybrid modeling structure would eliminate some of the manual labor needed. Upon using the hybrid simulation option the meso and micro level could be combined so as to reduce the effort needed in the micro model extraction as seen in Figure .

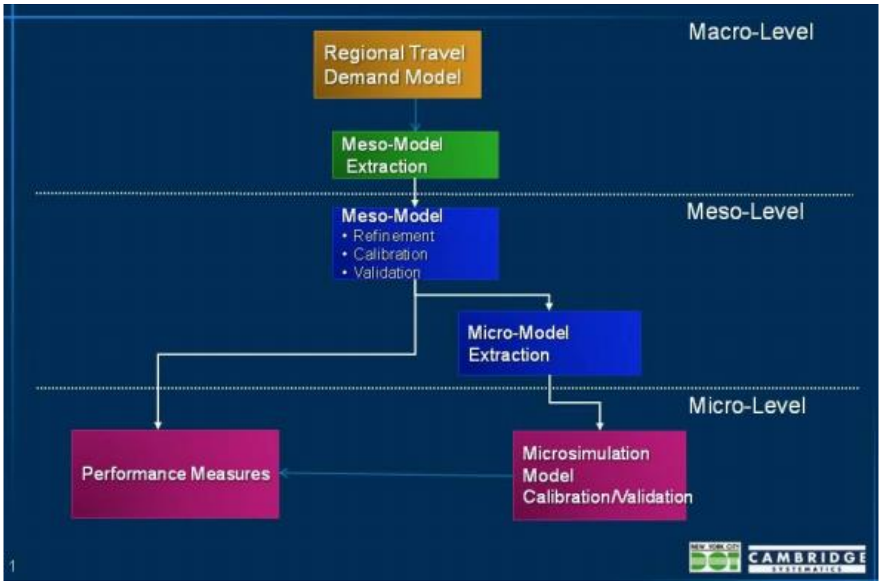


Figure 17: MTM full model run structure

Table 14: MTM DTA model summary

MANHATTAN TRAFFIC MODEL (MACRO, MESO AND MICRO)

ROAD LENGTH	1,627 km
LANE LENGTH	4,520 km
ZONES (TAZ)	1,583 centroids
NUMBER OF SECTIONS	9,720
NUMBER OF INTERSECTIONS	3,566
SIMULATION DURATION	from 5:30 to 10:00 (4h 30 min)
TOTAL DEMAND	1,437,542 vehicles
TIME TO SIMULATE	11 minutes (Time to complete 1 iteration as part of a DUE)
VALIDATION CRITERION	±15% difference between measured and simulated flows
CLIENT	NYC DOT
DATE OF COMPLETION	2011
COST	NA[31]

[31] Attempts were made to acquire the cost (both in time and money) but were not able to be acquired for this paper due to the reluctance of private companies and DOT’s to disclose or inability to locate such statistics.

Dynameq – San Francisco Dynamic Traffic Assignment Model

All information used in this section was derived from published reports (7, 8).

In 2010, the Federal Highway Administration (FHWA) gave a grant to the San Francisco County Transportation Authority to create a DTA model for the entire city of San Francisco. The project, referred to as “DTA Anyway”, is a continuation of a DTA model previously developed for the Northwest quadrant of San Francisco. Its main purpose for the local region is to develop a detailed model for projects in the region that cannot be effectively evaluated, if at all, by the San Francisco Chained Activity Modeling Process (SF-CHAMP).

It was also developed with the intention of being useful to other DTA models in other regions not in the immediate vicinity. Its main uses for other models include: building a flexible toolset for other models, documentation on the process and assumptions for developing a DTA model, and answering several challenging DTA questions. These included: how does DTA perform in a dense and highly congested grid network, how can DTA be used to study the interaction of the street network with the transit system, and what benefits might DTA provide in evaluating congestion pricing policies?

Geometry

San Francisco covers roughly 80 square miles surrounded on three sides by water and the fourth is partially blocked by the natural features of San Bruno Mountain State Park. This causes almost all traffic onto very well defined chokepoints such as the Golden Gate Bridge and the Bay Bridge. This allowed boundaries to be defined at these chokepoints to avoid “locking” entrances to the grid network at arbitrary streets, which was a notable issue in the smaller model previously developed for San Francisco. The road network therefore included every street in the city along with the 976 Traffic Analysis Zones. It also includes the 1,115 traffic signals and 3,726 other intersections that are not signalized but controlled with other methods. The entire DTA model can be seen in Figure .

The DTA model was constructed from the SF-CHAMP model that is currently maintained by the San Francisco County Transportation Authority. It is important to note that instead of using the macroscopic model once as a starting point, it was instead used as the base network. This is significant because by generating the network from SF-CHAMP each time the model is run it is guaranteed to “fit” or work within the demand model. This allows the demand model to be updated and allows the DTA network to natively accept the demand output from SF-CHAMP. This method required the development of several processes that have to be followed. The first involved creating an automated process for many aspects of the model such as preparing inputs, summarizing results, and converting data. The second involved fixing all items that cannot be automated, such as geometric changes, to the “source” or the SF-CHAMP model. A third was how the model was developed to allow simultaneous editing by multiple users. This is essential for any large network as having a single engineer/technician working on the model is often infeasible given model complexity and project time tables. By generating the model from scratch every time, the “script” that assembles the network can easily be edited using a version control system. In this case the Git[32] control system was used and allowed issues to be tracked and changes between model iterations to be easily monitored.

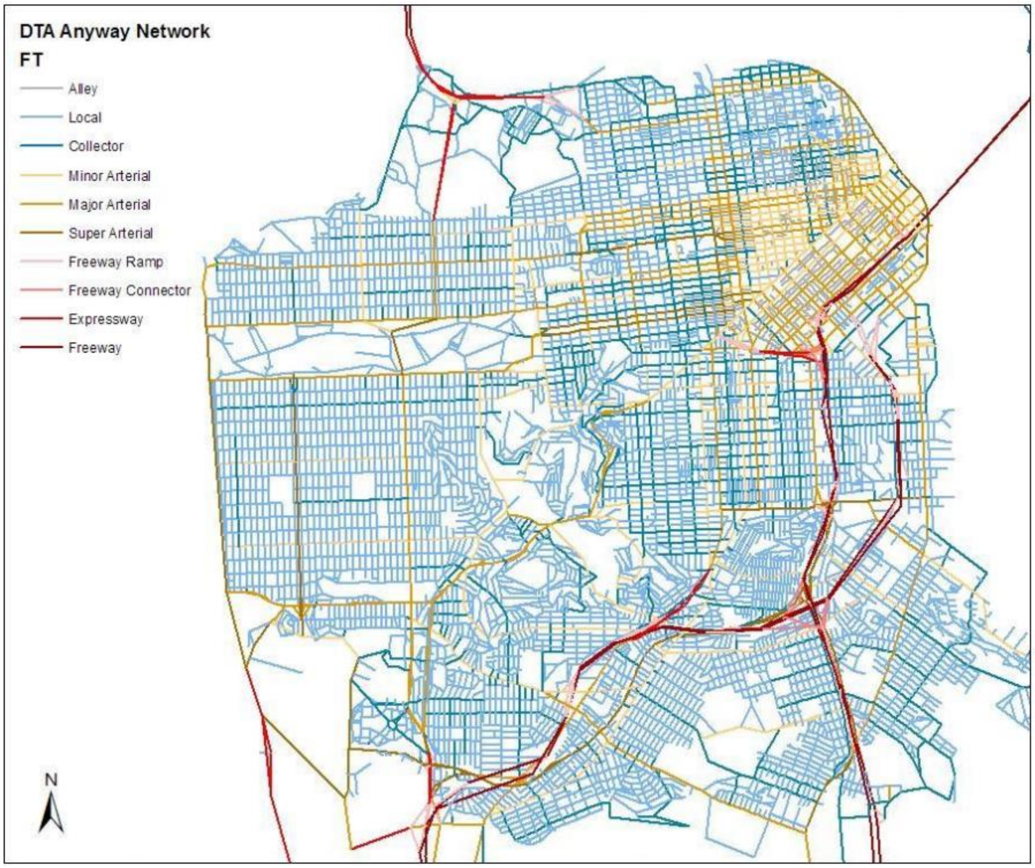


Figure 18: San Francisco DTA model by road class

Control

The San Francisco Municipal Transportation Agency (SFMTA) maintains, in an Excel format, every signal timing card for every signalized intersection in San Francisco (around 1,115 in total). Every signal card contains the complete cycle length, phase, and coordination offset as they change throughout the day. They also contain how actuated signals operated, if they were actuated, in addition to a default fixed

[32] Git is a distributed revision control system often used by programmers working on the same project. http://git-scm.com/

timing plan. Upon examination of the signal timing cards however it was discovered that many of the older signal timing plans did not follow the same standardized structure of more recent plans. The signal cards were therefore systematically standardized into the most recent format so they could be imported into the model.

As was noted previously this DTA model is different from most in that it is generated from scratch each time. Since the signals were going to be implemented each time an automated process was developed to not only import each signal but to check the validity of the signal after it is imported to avoid the most common errors in programming signals. These error checks included ensuring every phase has at least some green time and identifying conflicting protected movements that are both given green simultaneously.

This method of importing signals not only reduces errors caused by programming the signals individually but allows the signals to be changed relatively quickly in the Excel file. Since all new signal timings follow the standardized format any changes to the signal network can be quickly implemented. The biggest downfall however is the fact each signal could only have one signal timing plan associated to it during a simulation period. Therefore if a signal timing changed midway during the simulation it could not be accounted for. This feature could be implemented in the code and is seen as a potential update in future extensions of the model.

Demand

Since the San Francisco DTA model is directly derived from the SF-CHAMP regional model it can directly incorporate the regional model's trip tables. A subarea extraction of the model at the chokepoints allowed for well-defined demand matrices to be extracted for the four user classes: autos, trucks, auto-toll, and truck-toll. Although no toll vehicles exist they are placeholders for future studies and represent the flexibility in the model in incorporating any class of vehicle. The current model is focused around the PM peak and simulates traffic from 2:30-7:30pm which includes both the warm up and cool down hours. In total around 535,200 cars and 84,200 trucks enter the network during the simulation period.

Calibration

During the calibration portion of the SF-DTA model significant effort went into collecting and ensuring that all inputs were of high enough quality to be useful and thus avoid a "garbage in garbage out" scenario. Notable effort went towards cleaning network geometry and ensuring that signal timings were accurate. The developers also sought to measure all aspects of reality that could be measured in order to calibrate the model to real world values. Those measurements were taken into high regard in order to avoid changing speeds and capacities during model calibration. Instead more effort was focused into calibrating the generalized cost equation.

Once a DTA was run the model was first looked over qualitatively to ensure that excessive gridlock did not skew the results. If excessive gridlock was found only changes relative to relieving the unrealistic gridlock were applied to the model to avoid changing the network because of problems resulting from the gridlock. Following the qualitative analysis the network was analyzed quantitatively using link/turning movement counts varying from 5-60 minutes taken from multiple sources ranging from 2009-2011 and further filtered to Tuesday-Thursday. The location of these counts can be seen in Figure which does not include the additional 15 freeway sections with 5 minute statistics covering 59 mainline lanes throughout the region modeled.

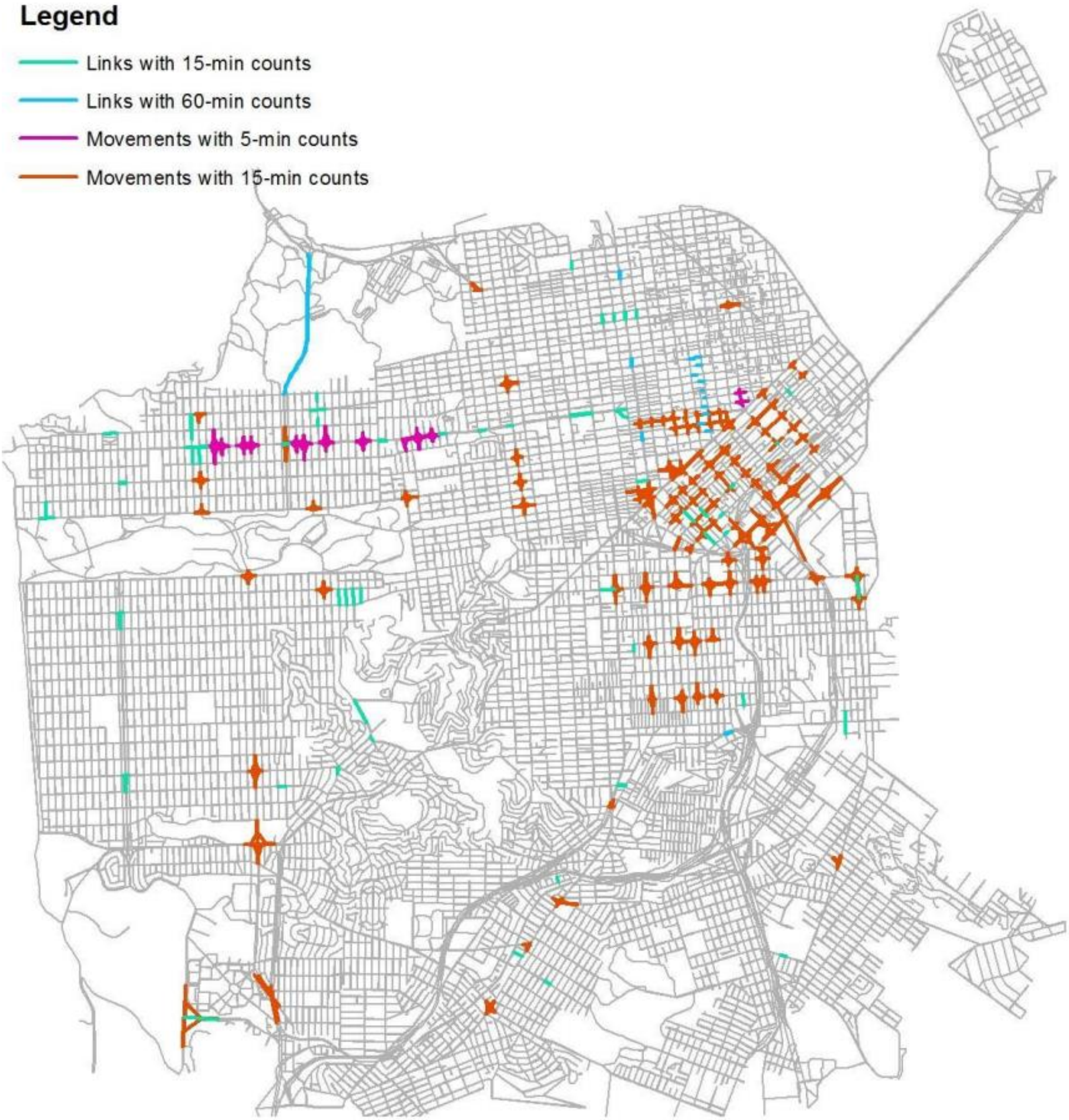


Figure 19: Calibration count locations for SF-DTA model

Once the SF-DTA Model was producing reasonable results without issues stemming from coding errors, the model was further calibrated by adjusting speed/flow parameters, generalized cost functions, pedestrian friction, transit-only lane permissions, signal phases, and demand loading. During this calibration the model was often found to be highly sensitive to certain parameters. For example the model was found to be sensitive to the length of trucks; altering the length by as little as a foot could often cause the network to go into complete gridlock. The programmers thus had to run multiple tests to find the correct truck length while maintaining a relationship between trucks and cars where trucks were about 1.5 times the length of cars.

Early results also appeared to show that demand on the major arterials and freeways was lower than field observations. The static model however showed a reasonable usage of those links with the exact same demand. The developers therefore concluded that the demand was not disappearing but was instead spreading over more links that did not have data available. They also found that vehicles would tend to zigzag through the network as well as use very small local streets that had free flow speeds. To solve these issues the generalized cost function in Dynameq was adjusted multiple times before converging on the finalized form shown below.

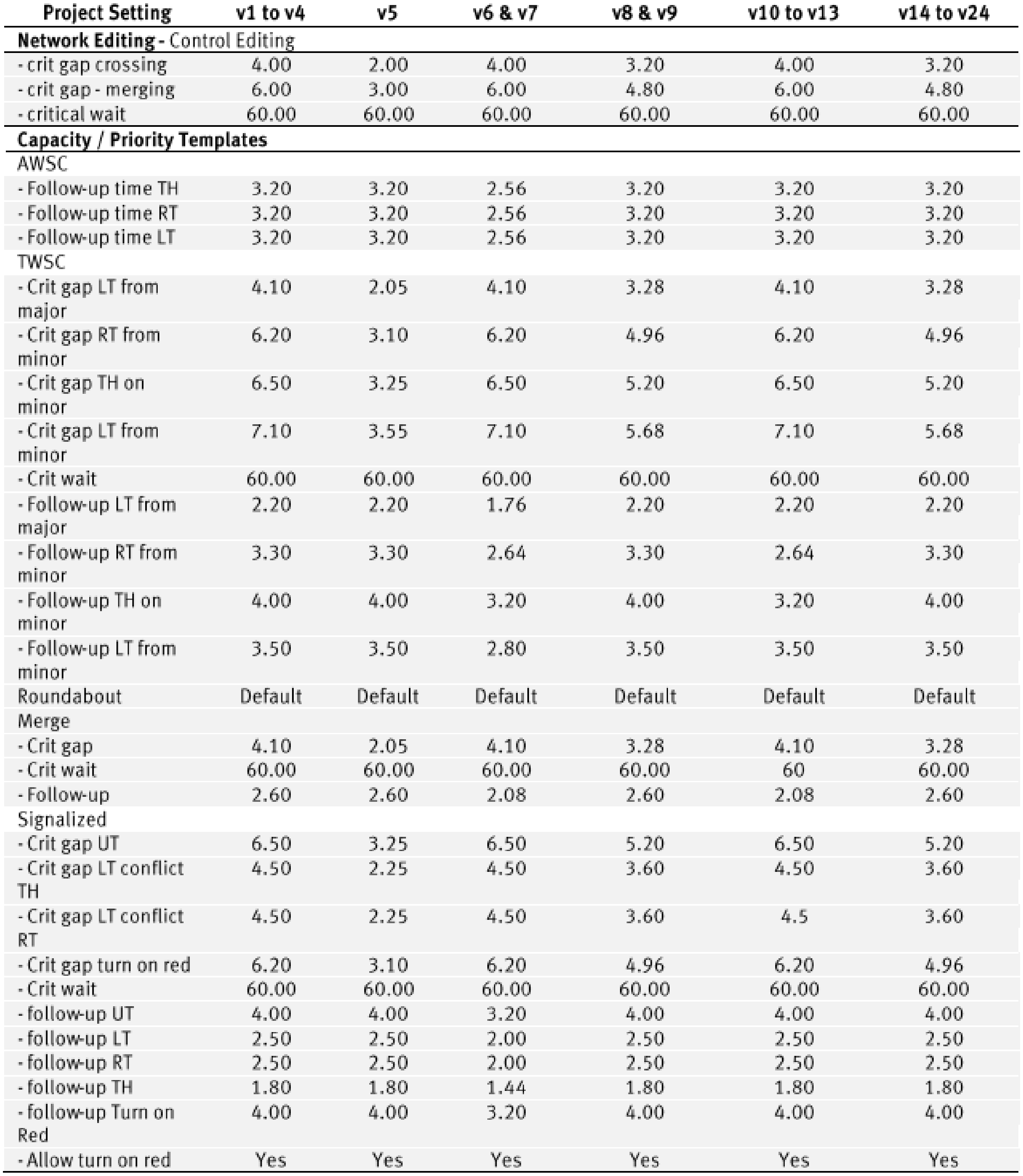
$$\begin{aligned} Generalized\ Cost \\ &= Time + LeftTurnPenalty + RightTurnPenalty + FacilityTypePenalty \\ &+ TollPenalty \end{aligned}$$

where:

$$\begin{aligned} &LeftTurnPenalty = 30\ seconds\ if\ a\ movement\ is\ a\ left\ turn \\ &RightTurnPenalty = 10\ seconds\ if\ a\ movement\ is\ a\ right\ turn \\ &FacilityTypePenalty = 50\%\ of\ free\ flow\ travel\ speed\ if\ link\ is\ alley, local\ or\ collector \\ &Toll\ Penalty = Toll\ as\ specified\ in\ the\ scenario \end{aligned}$$

In addition to modifying the generalized cost equation, many of the model's global parameters were altered to better replicate the driving behaviors of the local residents. The amount of parameters available can be seen in Table , which shows the settings for all-way stop controlled (AWSC), two-way stop controlled (TWSC), and signalized intersections.

Table 16: San Francisco DTA section parameters



Project Setting	v1 to v4	v5	v6 & v7	v8 & v9	v10 to v13	v14 to v24
Network Editing - Control Editing						
- crit gap crossing	4.00	2.00	4.00	3.20	4.00	3.20
- crit gap - merging	6.00	3.00	6.00	4.80	6.00	4.80
- critical wait	60.00	60.00	60.00	60.00	60.00	60.00
Capacity / Priority Templates						
AWSC						
- Follow-up time TH	3.20	3.20	2.56	3.20	3.20	3.20
- Follow-up time RT	3.20	3.20	2.56	3.20	3.20	3.20
- Follow-up time LT	3.20	3.20	2.56	3.20	3.20	3.20
TWSC						
- Crit gap LT from major	4.10	2.05	4.10	3.28	4.10	3.28
- Crit gap RT from minor	6.20	3.10	6.20	4.96	6.20	4.96
- Crit gap TH on minor	6.50	3.25	6.50	5.20	6.50	5.20
- Crit gap LT from minor	7.10	3.55	7.10	5.68	7.10	5.68
- Crit wait	60.00	60.00	60.00	60.00	60.00	60.00
- Follow-up LT from major	2.20	2.20	1.76	2.20	2.20	2.20
- Follow-up RT from minor	3.30	3.30	2.64	3.30	2.64	3.30
- Follow-up TH on minor	4.00	4.00	3.20	4.00	3.20	4.00
- Follow-up LT from minor	3.50	3.50	2.80	3.50	3.50	3.50
Roundabout	Default	Default	Default	Default	Default	Default
Merge						
- Crit gap	4.10	2.05	4.10	3.28	4.10	3.28
- Crit wait	60.00	60.00	60.00	60.00	60	60.00
- Follow-up	2.60	2.60	2.08	2.60	2.08	2.60
Signalized						
- Crit gap UT	6.50	3.25	6.50	5.20	6.50	5.20
- Crit gap LT conflict TH	4.50	2.25	4.50	3.60	4.50	3.60
- Crit gap LT conflict RT	4.50	2.25	4.50	3.60	4.5	3.60
- Crit gap turn on red	6.20	3.10	6.20	4.96	6.20	4.96
- Crit wait	60.00	60.00	60.00	60.00	60.00	60.00
- follow-up UT	4.00	4.00	3.20	4.00	4.00	4.00
- follow-up LT	2.50	2.50	2.00	2.50	2.50	2.50
- follow-up RT	2.50	2.50	2.00	2.50	2.50	2.50
- follow-up TH	1.80	1.80	1.44	1.80	1.80	1.80
- follow-up Turn on Red	4.00	4.00	3.20	4.00	4.00	4.00
- Allow turn on red	Yes	Yes	Yes	Yes	Yes	Yes

Overall the model was calibrated until 65% of all links fell within an acceptable range as seen in Figure according to the Caltrans Travel Forecasting Guidelines[33]. This was in conjunction with 47% of all movements falling in those same criteria where the low target is the line closer to the observed axis and the high closer to the modeled axis.

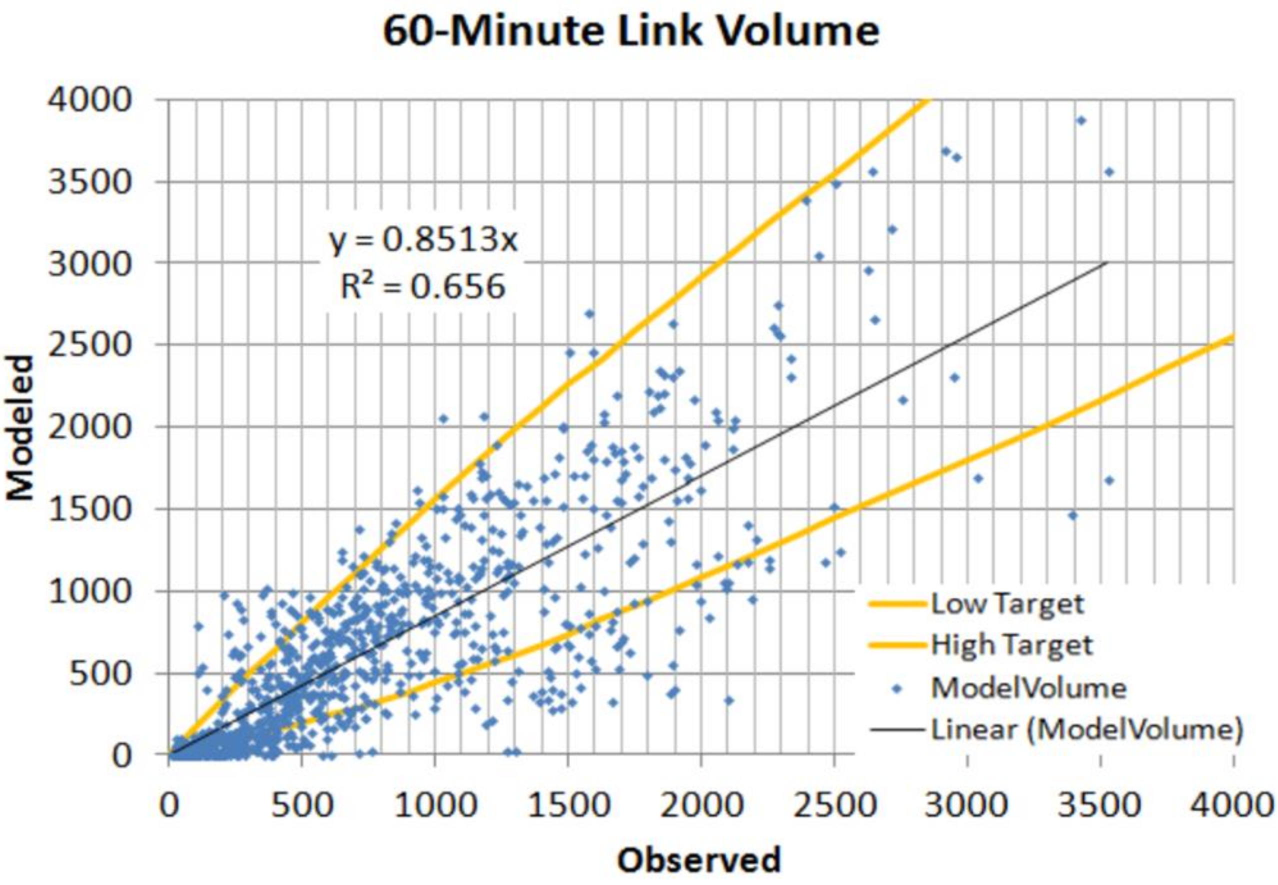


Figure 20: SF-DTA link volumes modeled vs observed

The Ohio RMSE Curve was also used as a guideline for validation purposes. It can be seen in Figure that modeled RMSE values were below the Ohio RMSE target values by 10% or more. This was also supported by the convergence of the model at around 100 iterations as seen in Figure . Across all time periods the model had an average R-Gap of 2.7% which was below the self-imposed target of 5% and more importantly it is visually stable.

[33] State of California Department of Transportation Travel Forecasting Guidelines, November 1992

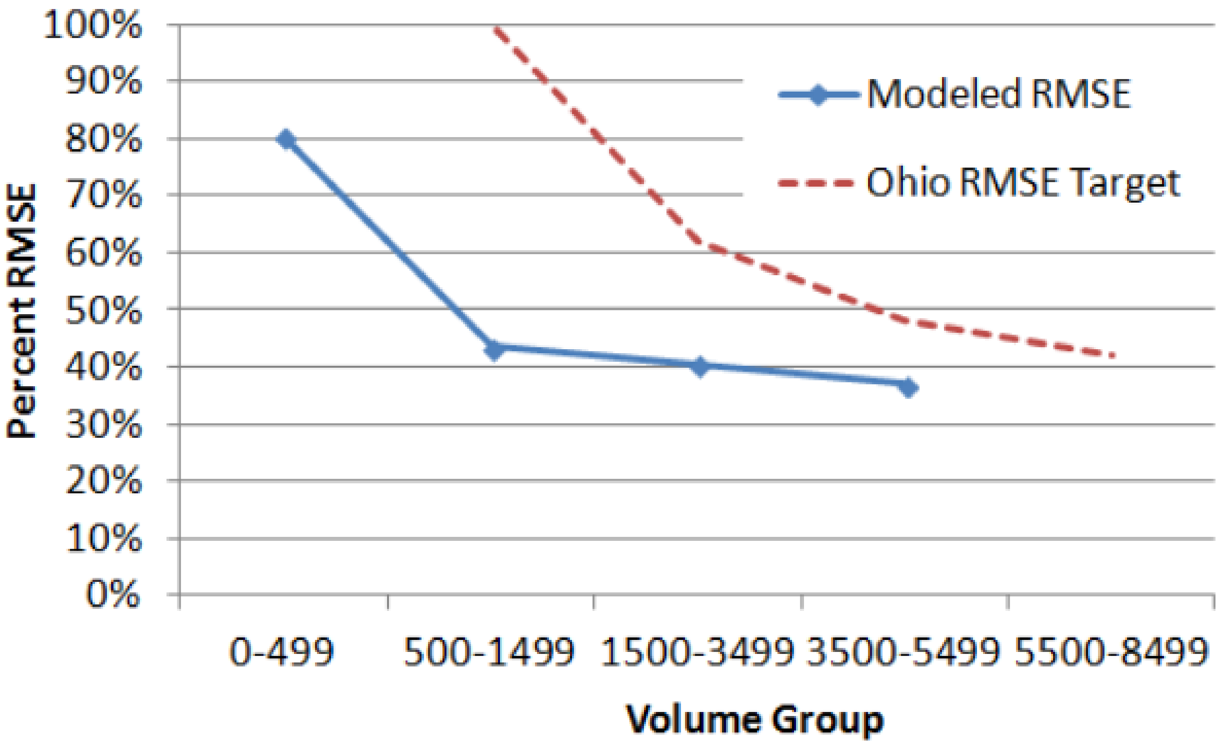


Figure 21: San Francisco RMSE by volume group

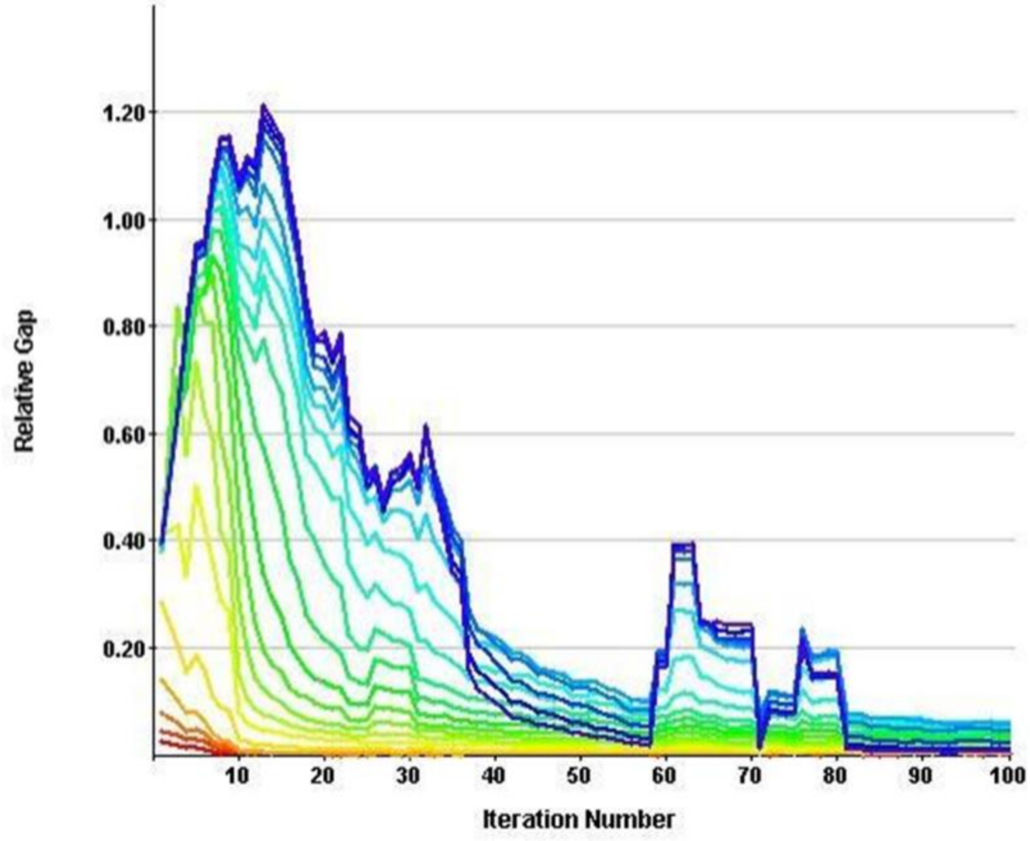


Figure 22: Convergence graph for San Francisco base scenario

Summary

This project provided significant insight into how a DTA network could be developed. Figure shows how all of the different parts of the model are assembled and how the different pieces can be added, replaced, modified, or removed. It appears through these reports to be a viable option for maintaining consistency among multiple researchers as well as giving full control over the network to run more complicated experiments such as sensitivity analysis. The down side to this type of system is that a strict form of documentation is needed to explain the association between files so that it can be used again by others who were not involved in the initial development. This documentation can be further complicated since most traffic simulation suites rely on their respective GUI's to make these associations and to maintain cohesiveness between all the files. A summary of essential model statistics was provided in

Table and

Table to allow a quick comparison between the different models.

Table 17: San Francisco DTA model summary

SAN FRANCISCO DTA MODEL

ZONES (TAZ)	976 TAZs, 22 external stations
NUMBER OF SECTIONS	37,000
NUMBER OF INTERSECTIONS	1,115 signalized, 3,726 stop controlled (15,000 nodes)
SIMULATION DURATION	2:30 to 7:30 (5hr), including 1hr warmup/cool down
TOTAL DEMAND	535,200 cars, 84,200 trucks
TIME TO SIMULATE	52.5 hours
VALIDATION CRITERION	Caltrans Travel Forecasting Guidelines
DATE OF COMPLETION	2012
CLIENT	San Francisco County Transportation Authority
COST	NA[34]

[34] Attempts were made to acquire the cost (both in time and money) but were not able to be acquired for this paper due to the reluctance of private companies and DOT's to disclose or inability to locate such statistics.

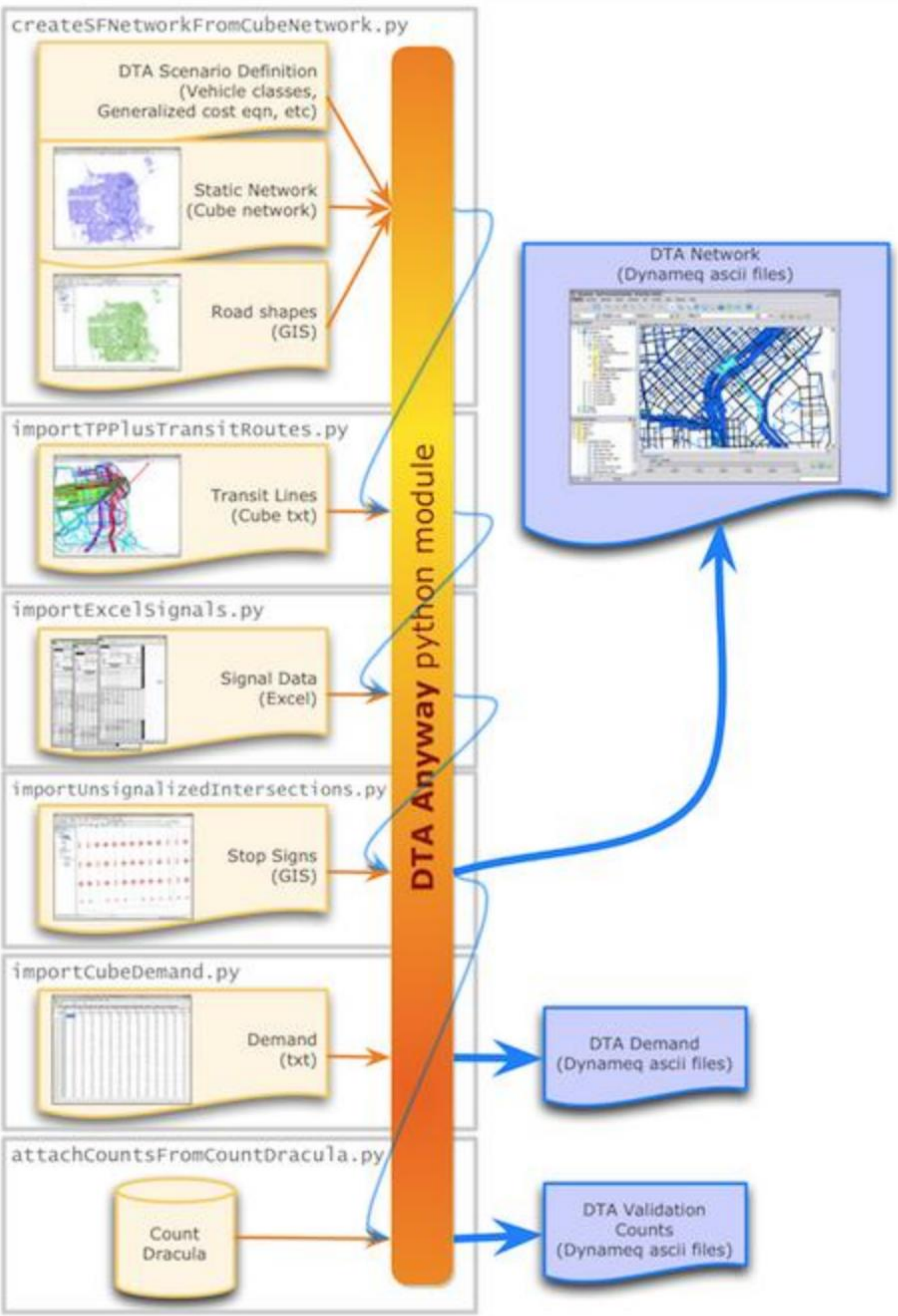

createSFNetworkFromCubeNetwork.py
DTA Scenario Definition (Vehicle classes, Generalized cost eqn, etc)
Static Network (Cube network)
Road shapes (GIS)
importTPPlusTransitRoutes.py
Transit Lines (Cube txt)
importExcelSignals.py
Signal Data (Excel)
importUnsignalizedIntersections.py
Stop Signs (GIS)
importCubeDemand.py
Demand (txt)
attachCountsFromCountDracula.py
Count Dracula
DTA Anyway python module
DTA Network (Dynameq ascii files)
DTA Demand (Dynameq ascii files)
DTA Validation Counts (Dynameq ascii files)

Figure 23: Flow chart of script sequence used in the SF-DTA model

DynusT – Detroit I-96 freeway DTA Model

All information used in this section was derived from an unpublished report (9) and a Transportation Research Board paper (10).

The Detroit DTA model is a large Meso-DTA model that was designed to analyze the impacts of various construction staging strategies for the reconstruction of I-96 between I-275 and US-24 produced by Parsons Brinckerhoff, Inc. (PB).

Geometry

The Southeast Michigan Council of Governments (SEMCOG) regional macroscopic travel demand forecasting model (TDFM) uses a typical four-step model structure and was used as a starting point for the model. Since the TDFM was constructed in TransCAD[35] and is a GIS network in general it could be used directly as it was to extract a sub area and improve it to a level suitable for DTA simulation. The DTA subarea model covering roughly 700 square miles includes all alternative roadways that would be impacted by construction of I-96 as seen in Figure . Besides the original deletion of links not in the DTA subarea no other links were added or altered from the TDFM Model.

[35] http://www.semcog.org/TravelForecast_TDFM.aspx

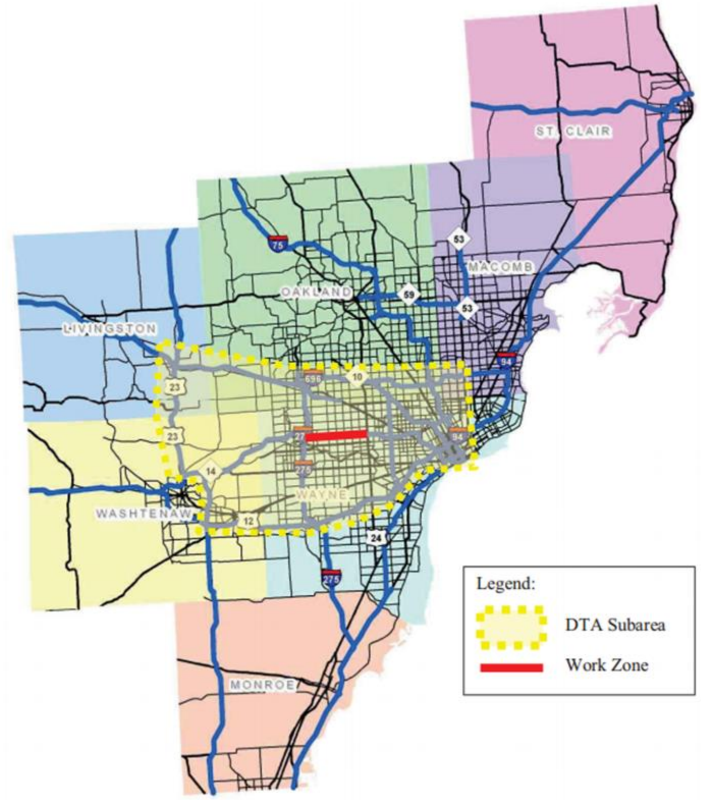


Figure 24: Map of the Detroit regional and Detroit-DTA subarea

Control

The Detroit DTA model does not appear, through the available literature, to have any form of traffic control information incorporated in it. As users of the regional planning model here in the Twin Cities it is our opinion that signal information was most likely incorporated through the section attributes. As regional travel demand forecast models are almost always static they cannot incorporate traffic control delay into the assignment. Therefore they often take into account the signal control delay by modifying the section or link parameters such as maximum speed and capacity. Given that the Detroit I-96 freeway DTA Model was directly extracted from the regional travel demand forecast model this appears to be a reasonable assumption for how the control was implemented.

Demand Calibration/Validation

Due to the way this particular model was developed the demand and calibration portion became blended and could not be easily discussed separately. Therefore this section discusses both steps.

Before the subarea model was extracted and calibrated a dynamic demand adjustment was done on the regional DTA model after it was found that the demand produced by the TDFM for the subarea model was not calibrated sufficiently to be used as it was. As can be seen in Figure the original origin/destination demand matrices produced by the TDFM (in TransCAD), converted to DynusT format, and finally used by the regional DTA model (in DynusT) did not match the observed counts and thus would provide unsuitable inputs for the smaller DTA model.

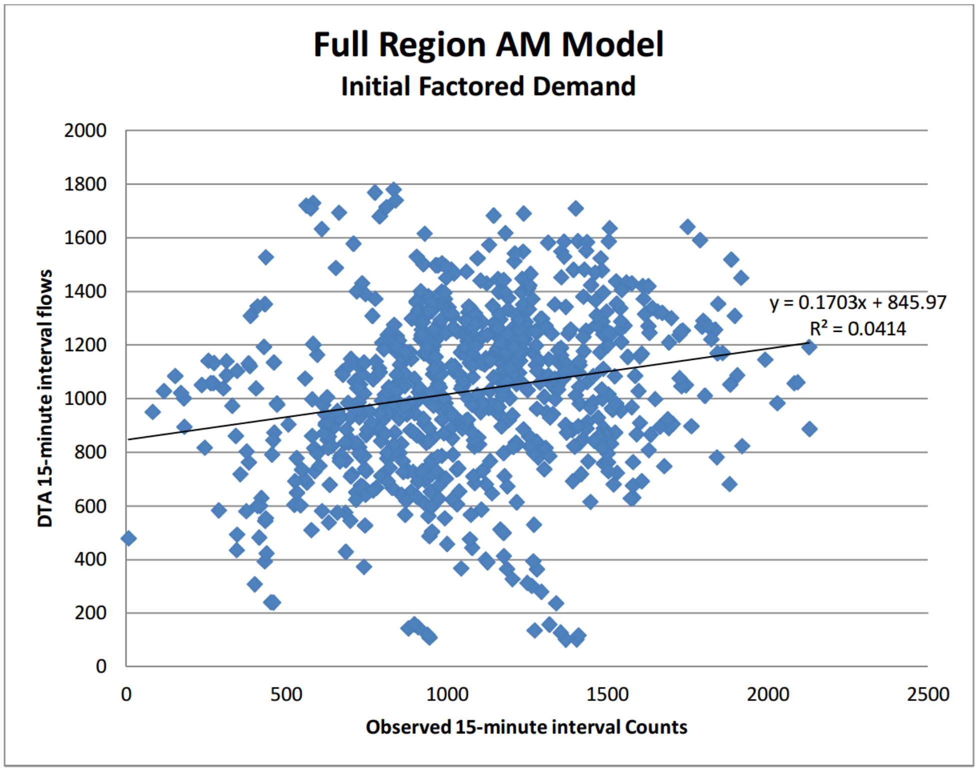


Figure 25: Initial full region demand for the Detroit-DTA model

Given the large number of data points an automated process was developed since a manual adjustment would have been infeasible given the size of the network. In order for this to work, however, real data with which to adjust would be needed. The developers were able to use 15 minute traffic volumes from Traffic.com[36] from every Tuesday, Wednesday, and Thursday during October 2010 to assemble the real data. This demand data was then used to perform the dynamic demand adjustments on both the regional and smaller DTA model as follows:

1. Initial DTA model (one-shot) is run to produce a vehicle trajectory file.
2. A balancing procedure is run to determine vehicle factors which, when summarized as link flows, closely replicate associated link counts. This procedure changes the demand patterns in the input O/D matrixes.

[36] Traffic.com was acquired by Navteq in 2006 and in 2013 Nokia (Navteqs parent company) merged traffic.com with its HERE mapping division. Traffic.com is now located at www.here.com/traffic

3. Vehicle and path input files are prepared for the DTA model.
4. A subsequent DTA model run (DUE) is created to produce a new, improved vehicle trajectory file.
5. Steps 2-4 are repeated until satisfactory adjusted demand is achieved.

As was mentioned earlier, fully calibrating and running all the senarios for the full regional model was infeasible both computationally and due to time constraints. However the regional travel demand model was able to be run “enough” to produce far better demand inputs for the smaller DTA model. Figure shows how they were able to use this dynamic adjustment technique to improve the initialinitial demand seen in Figure .

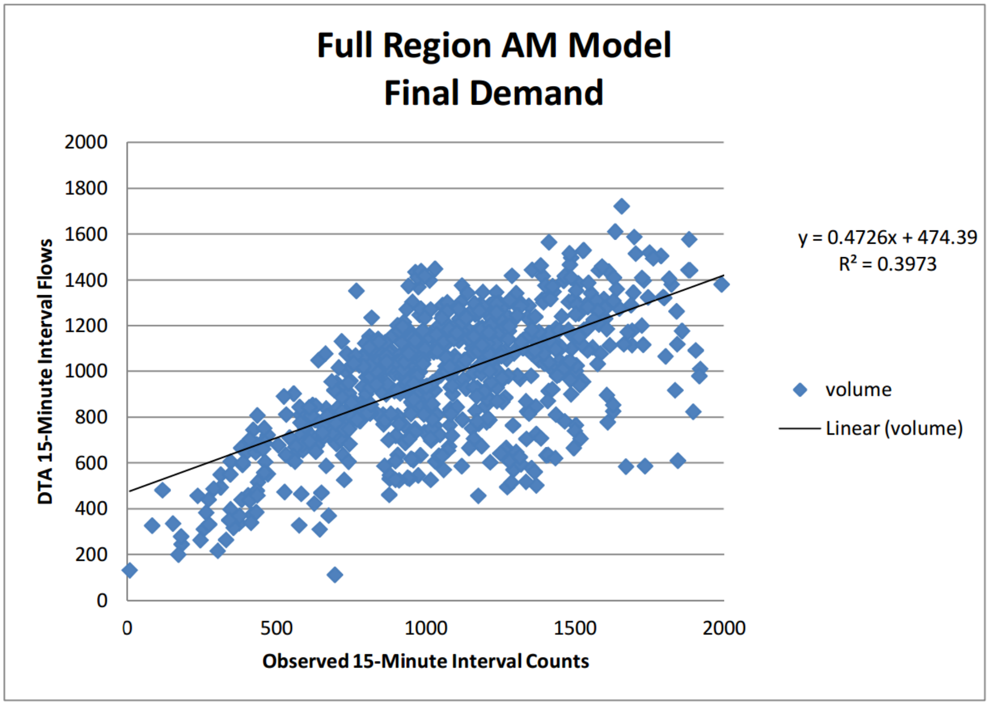


Figure 26: Scatter plot of morning peak partially calibrated regional demand model flows vs observed counts

With the regional model partially calibrated the developers extracted the subarea model. During the extraction the origins and destinations within the subarea maintained their identities from the larger regional travel demand model. However the border links that were attached to the network now had to have new Origin/Destination centroids that were synthesized from the vehicle trajectory files of the full regional model run.

The same dynamic demand adjustment procedure described above was then done for the subarea DTA model and after around 4 iterations of the full loop the demand had been refined to a high enough quality, as seen in Figure , for the developers to begin running the closure scenarios.

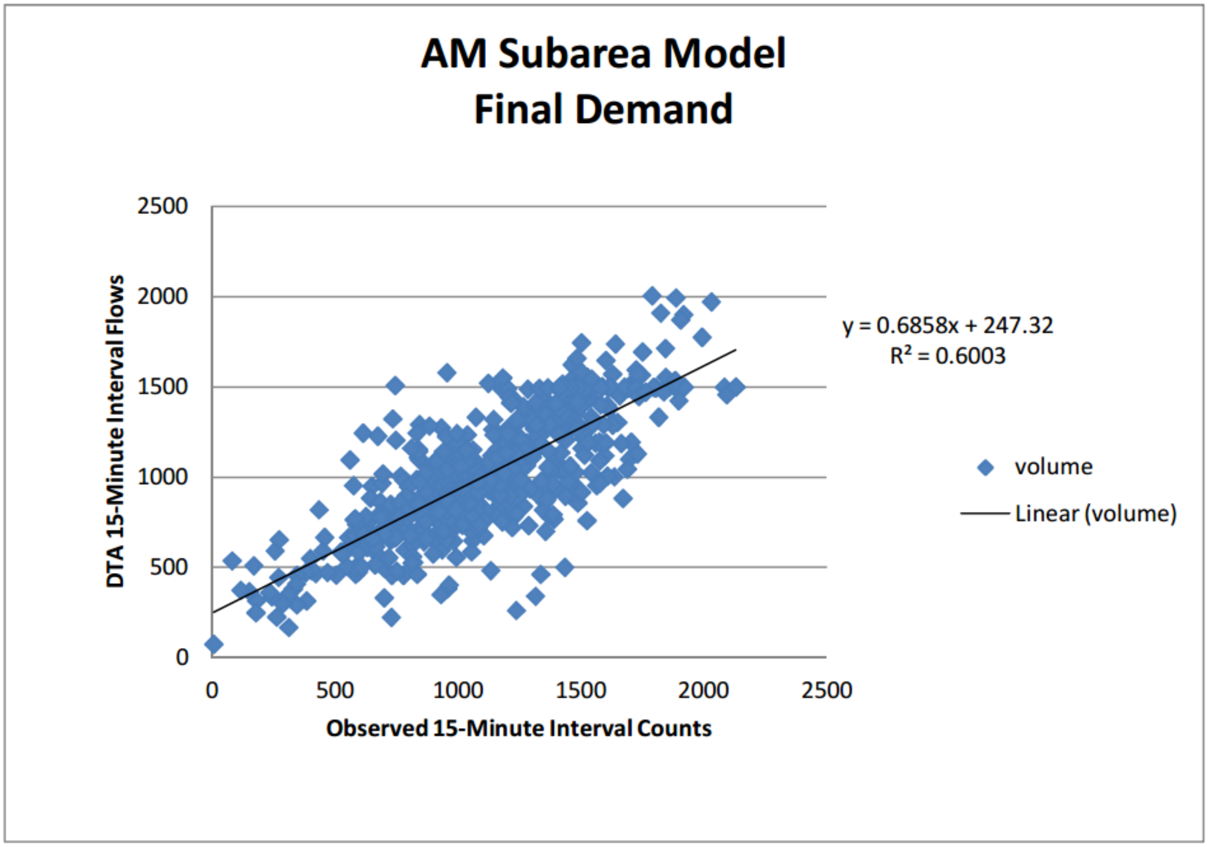


Figure 27: Scatter plot of morning peak calibrated demand for subarea DTA model flows vs observed counts

The model was also found to converge with their R-Gap values consistently being below 1% after around 30-35 iterations of the DUE. They also took into account the number of vehicles that switched routes and how much the average travel time was changing to decide when the model had converged. The amount of vehicles switching routes was around 3.5% and based off the guidance of colleagues from the University of Arizona this was found to be more than adequate. The value of average travel time was changing by only hundredths of a minute towards the end of the simulation. All of these measurements were used to determine when the model had converged on a solution.

Summary

Throughout the literature available on this project it is evident in our opinion that a few key details are missing. The first is how they dealt with the traffic signals in the area. Since this is a DTA model and many alternative routes are available to each vehicle they are likely to use arterials that have traffic control. There does not appear to be any mention of how the delays at these signals are incorporated into the model but as was discussed in the control section above they may have incorporated them in the section parameters. The second has to do with how, if needed, they dealt with multilane freeways where only single lanes break down, weaving, or other situations occur that would affect the usage of the section. Since DynusT does not recognize lane level decisions it cannot simulate congestion or delay caused by these characteristics. A summary of essential model statistics was provided in Table to allow a quick comparison between the different models.

Table 19: Detroit DTA model summary

DETROIT DTA MODEL

ZONES (TAZ)	~1200 TAZ
NUMBER OF SECTIONS	~20,000 (links)
NUMBER OF INTERSECTIONS	NA
SIMULATION DURATION	AM Peak 4hr (6:00-10:00) PM Peak 5hr (2:00-7:00)
TOTAL DEMAND	NA
TIME TO SIMULATE	NA
VALIDATION CRITERION	15 minute traffic counts from traffic.com (# of locations NA)
DATE OF COMPLETION	2013
COST	NA[37]

[37] Attempts were made to acquire the cost (both in time and money) but were not able to be acquired for this paper due to the reluctance of private companies and DOT's to disclose or inability to locate such statistics.]

TransModeler – Jacksonville DTA Model

All information used in this section was derived from unpublished internal report (11).

The TransModeler DTA Model of Jacksonville FL was created by the Caliper Corporation for the North Florida Transportation Planning Organization (NFTPO) and HNTB to cover the NFTPO six-county service area. This model has many of the same goals, in our opinion, as those here in the Twin Cities. The model was designed around extending the current regional activity-based and trip-based travel demand model, referred to as the Northeast Regional Planning Model (NERPM), and to provide a more operationally sensitive model. It is also seen as a potential replacement for the static traffic assignment portion of the NERPM.

An important distinction to note early is that this is a Microscopic DTA model. As a microscopic model it incorporates a higher-fidelity representation of driver behavior and signal operation with detail down to the lane-level. It includes every street in the Northeast Regional Planning Model as well as additional streets that extended the networks coverage down to the parcel level so it could directly take DAYSIM trips, a resident activity based travel demand model, which will be discussed further below in the demand section.

Geometry

This DTA model was created almost completely from scratch using aerial imagery. The only parts imported were the centroids and centroid connectors to ensure consistency between the different models. This was done so the DTA model could be interfaced loosely with the NERPM since they were built in different software packages. The main link between the two is the origins and destinations (OD's) as they are all that are needed to send skim matrices to NERPM and DAYSIM, and for the DTA to receive trip OD's from those demand models. As was mentioned earlier additional streets were added so that every parcel represented had a reasonable route from which to enter and exit the network. Very little detail was spared in this whole process as seen in Figure which represents an intricate geometry case encountered. The whole extent of the network can be seen in Figure covering around 6,000 square miles.

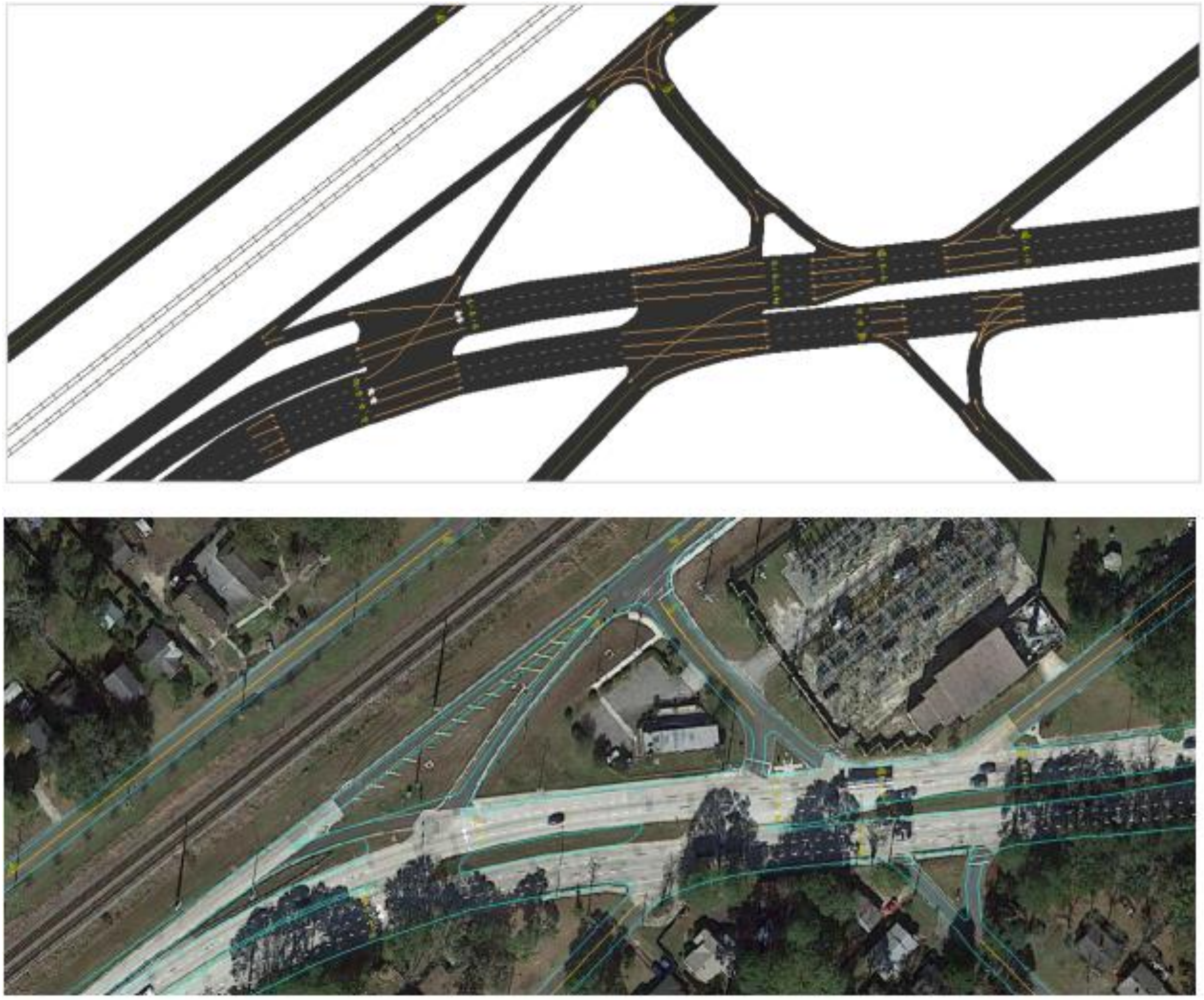


Figure 28: TransModeler intersection detail

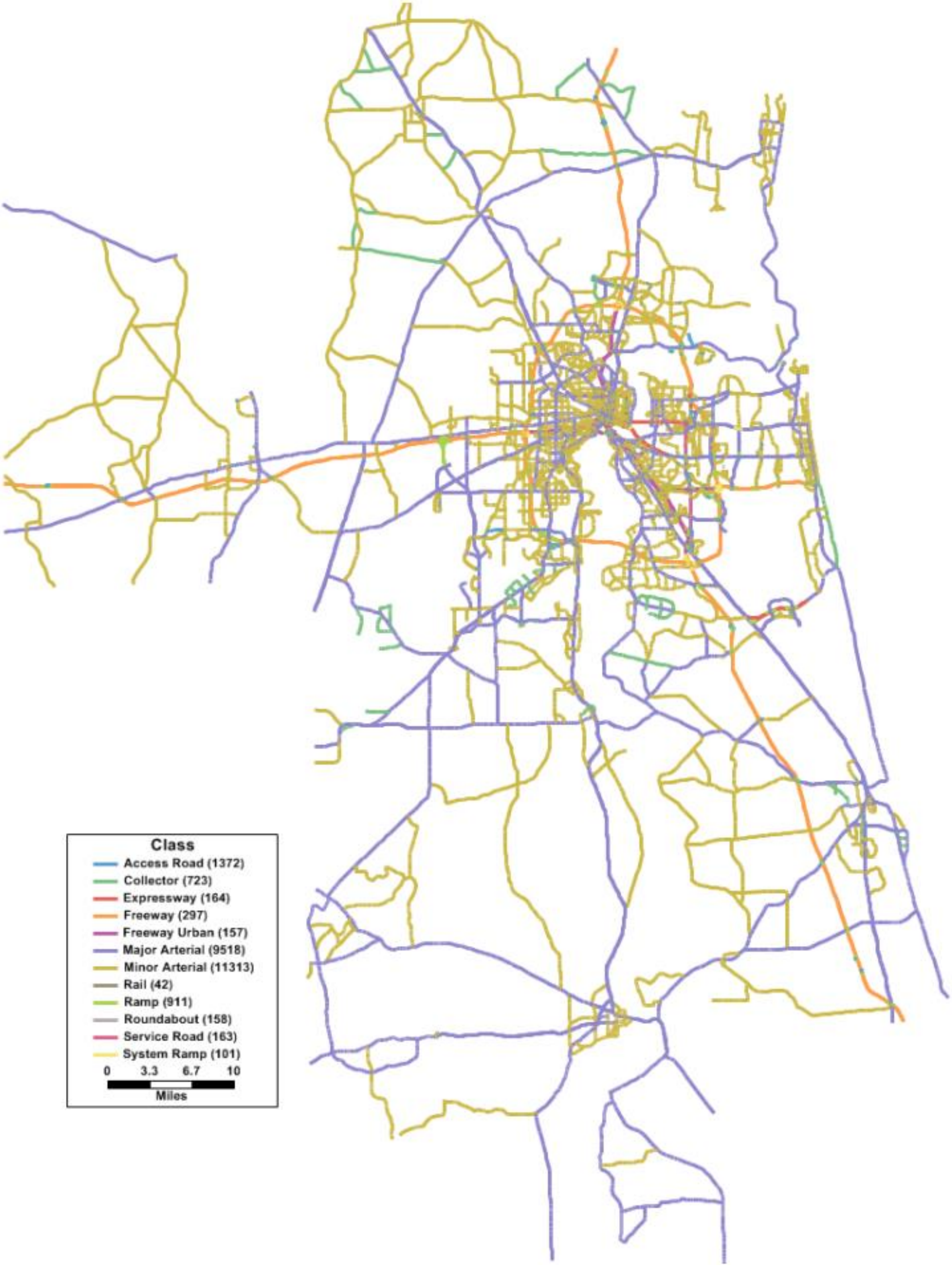


Figure 29: Jacksonville DTA network colored by road class

Control

Throughout the network there were approximately 1300 signalized intersections that need to be programmed. Since the traffic timing plans could not be inexpensively assembled a sample of signal plans was used to synthesize a basic style and design that would be used in the model. Many of the plans obtained were missing elements such as yellow and all-red times. Due to these issues, Google Street View was used to locate protected-permitted turns and default locations for detectors were assumed. With these assumptions the developers used TransModeler's signal optimization methods to develop signal timing that minimized the average delay throughout the network. Based our team's experience this method would not capture the effects of poorly timed signals and/or the coordination of signals across arterials. While this method is good for developing plans for intersections where no control information could be obtained, it is our opinion that key intersections or corridors should have more accurate signal timing to match the real world and not aggregated into an optimal solution.

Demand

As was mentioned earlier the DTA model was designed to be loosely integrated into the NERPM and use the demand generated by it as inputs. The NERPM was developed in the Cube architecture, using both the built in functions of Cube for truck forecasting and DAYSIM (implemented within Cube) for residential trip forecasting. DAYSIM accounts for all the aspects of residential trips and synthesizes a population to simulate a detailed itinerary for each person. The model incorporates aspects of travel behavior such as trip chaining, auto ownership/sharing, transit availability, and time of day scheduling down to the parcel level. Since the link between the NERPM and the Jacksonville-DTA model was based solely on the origins and destinations, which represent the parcels or TAZ's of the region, the centroids were the only requirement for the two models to be able to initiate a feedback loop between them and for the Jacksonville -DTA model to get the demand needed in order for it to run. This demand was then combined into three scenarios for the model: an AM Peak (5:00 – 9:00 AM), a MD Peak (9:00 AM – 3:30 PM), and a PM Peak (3:30 – 6:30 PM).

Because the NERPM is a macroscopic model, certain normal methods for loading demand into the network do not necessarily work well for a microscopic model. Often macroscopic models will load the network at a small number of arbitrary points around a given centroid such as intersections on adjacent arterials. Since vehicles do not load directly into intersections in reality and are instead fed onto arterials from collector streets, small portions of these streets were added so that traffic could load in a more realistic fashion. This also increased the number of centroid connectors for many centroids so that the demand for a particular TAZ was spread more uniformly around it instead of at a few isolated spots. This level of detail was often extensive and a sample can be seen in Figure .

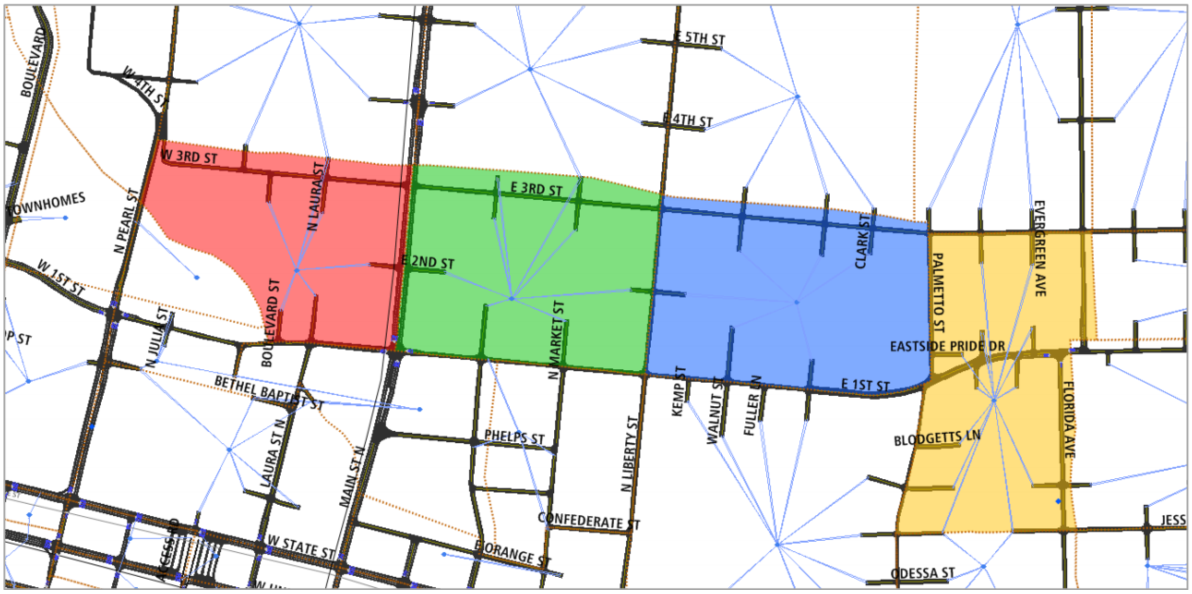


Figure 30: Jacksonville-DTA example illustration centroid connectivity in the model

The developers also did take note that the temporal shift of demand, or peak spreading, would be useful to capture and thus conducted experiments in an attempt to alter the demand produced by the NERPM to better fit the traffic count data. These experiments did not have the desired effects and only served to further isolate the relationship established between the NERPM and the DTA model. Since one of the main outcomes of the DTA model was a simplified interaction between the two the developers chose to omit the temporal shift and use the demand produced by the NERPM without alteration.

Calibration

The developers performed a high-level calibration/validation of the model that aimed at making sure the DTA model captured the regional traffic patterns and bottlenecks as well as confirm that the model was properly coded and the signal timing assumptions used and centroid connectivity were reasonable. The DTA model was a microsimulation model that was run iteratively for each simulation period so reasonable routes could be chosen based on congestion and travel times in a loaded network. To determine how much the model was changing between each iteration the method of successive averages was applied to the output travel time and turning delays. This method is also known as User Equilibrium (UE) and is run until a targeted Relative Gap is achieved or the maximum number of iterations is achieved. However since the simulation is based on a stochastic Monte Carlo simulation, every simulation has a different random seed which introduces its own amount of variability. In addition all trips were forced to integer values, and thus methods that allow static models to create fractional trips to better converge cannot be achieved in the microscopic DTA. When started from free flow conditions the model was found to converge in around 40-50 iteration with a relative gap of less than 1%, whereas when started with a previous solution it could achieve the same relative gap after around 20-30 iterations.

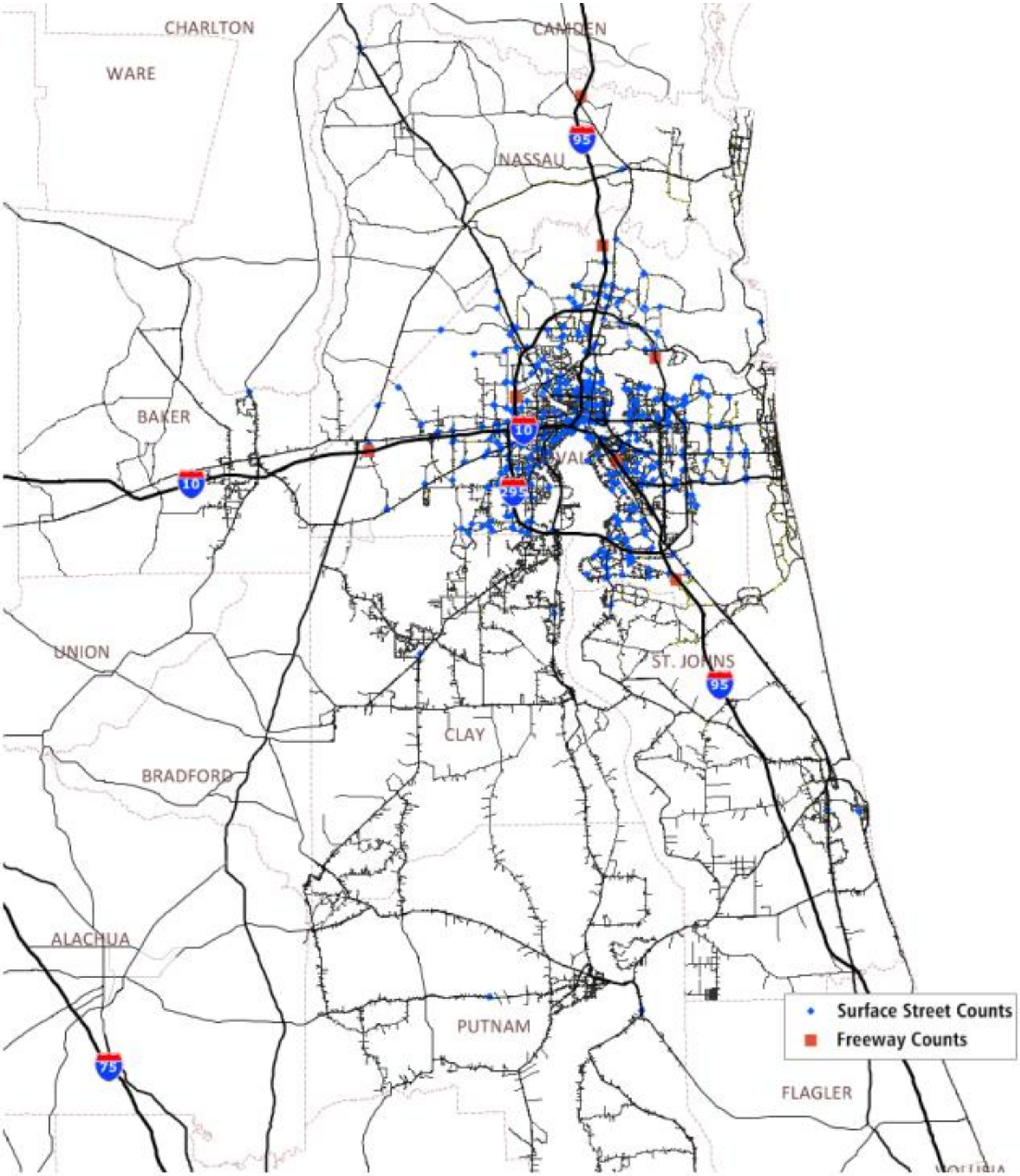
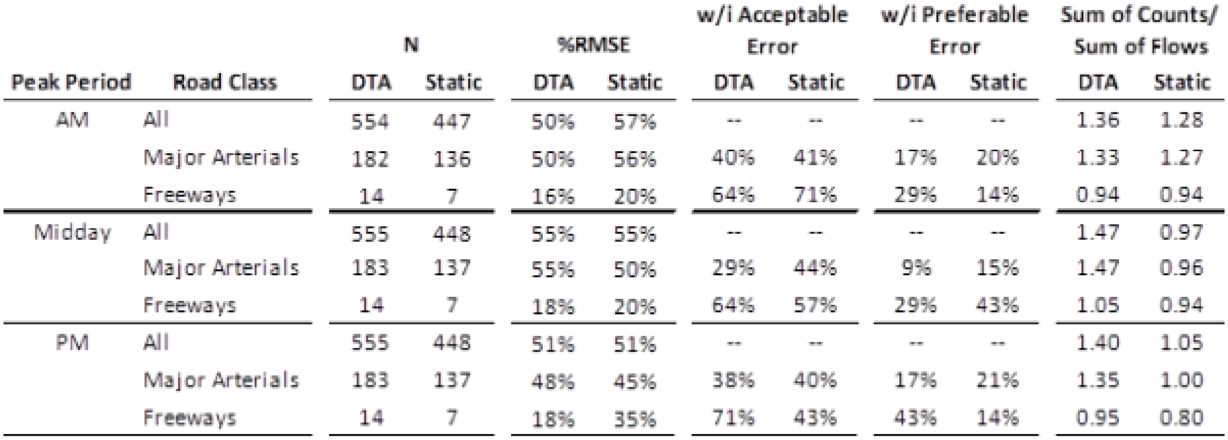


Figure 31: Traffic count locations for Jacksonville DTA model calibration

Using 555 traffic count locations, seen in Figure , the Jacksonville model was able to be calibrated to perform as well and/or better than the static traffic assignment model in the NERPM. This is seen in

Figure where in general the DTA model has a lower Percent RMSE as well as modeling freeway usage better than the static model. Through all of their graphs, such as Figure that showed counts vs. real data it was evident that in general there were fewer vehicles in the network than the real data would suggest. This in our opinion could have resulted from the over use of minor arterials in and around the network where network counts are difficult to obtain and thus requires local knowledge to know whether street assigned volumes are reasonable.



		N		%RMSE		w/i Acceptable Error		w/i Preferable Error		Sum of Counts/ Sum of Flows	
Peak Period	Road Class	DTA	Static	DTA	Static	DTA	Static	DTA	Static	DTA	Static
AM	All	554	447	50%	57%	--	--	--	--	1.36	1.28
	Major Arterials	182	136	50%	56%	40%	41%	17%	20%	1.33	1.27
	Freeways	14	7	16%	20%	64%	71%	29%	14%	0.94	0.94
Midday	All	555	448	55%	55%	--	--	--	--	1.47	0.97
	Major Arterials	183	137	55%	50%	29%	44%	9%	15%	1.47	0.96
	Freeways	14	7	18%	20%	64%	57%	29%	43%	1.05	0.94
PM	All	555	448	51%	51%	--	--	--	--	1.40	1.05
	Major Arterials	183	137	48%	45%	38%	40%	17%	21%	1.35	1.00
	Freeways	14	7	18%	35%	71%	43%	43%	14%	0.95	0.80

Figure 32: Percent RMSE and percent error threshold summary

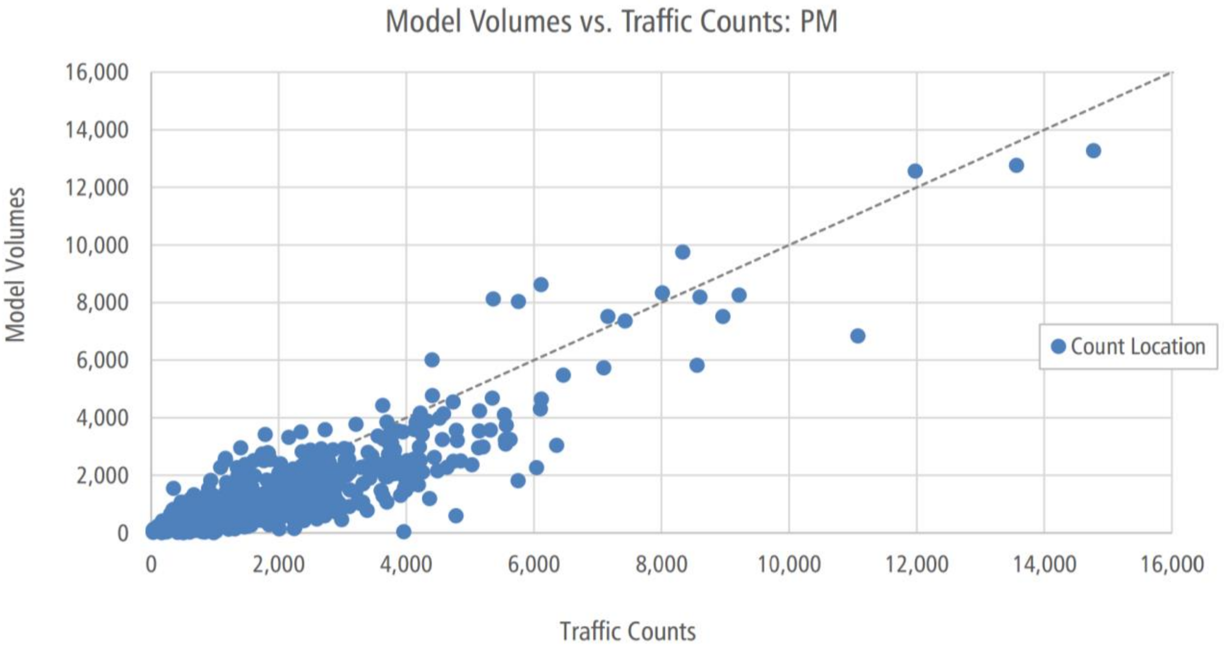


Figure 33: Scatterplot comparing PM DTA model volumes and traffic counts on all roads.

Summary

The Jacksonville DTA model closely relates to the structure of how a model could be implemented in the Twin Cities in terms of how its regional demand model is programed. Like the NERPM the Twin Cities Region Planning Model is based in CUBE and the underlining desire to replace the static traffic assignment in each is the same. The model however does have some downfalls such as the fact that actual signal timings were not used and instead an optimized solution was used that may impact how certain arterials operate. Since this is a high level calibration it is hard to say whether all the key bottlenecks or features of the area were accurately captured in the model. A summary of essential model statistics was provided in Table to allow a quick comparison between the different models.

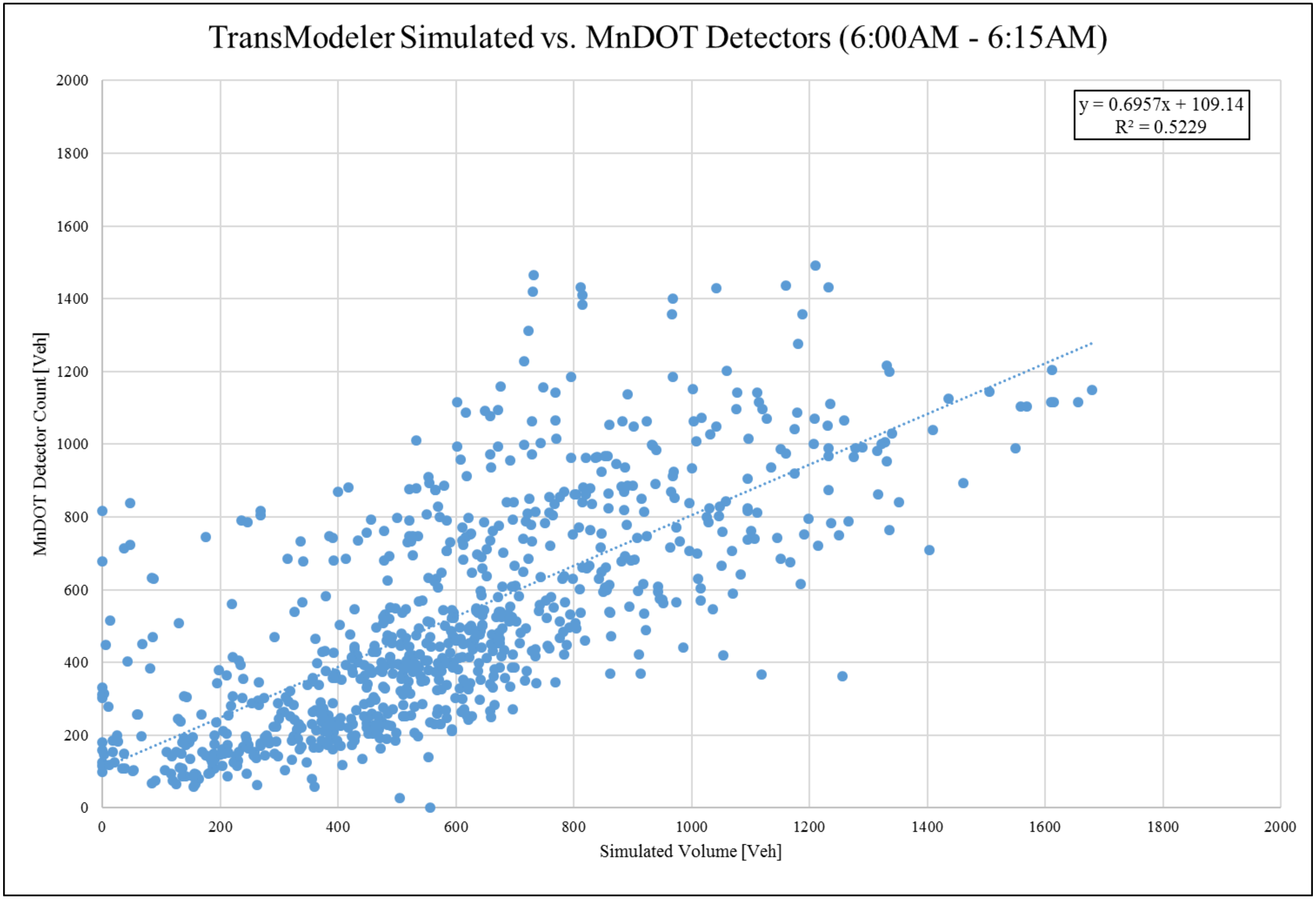
Table 20: Jacksonville DTA model summary

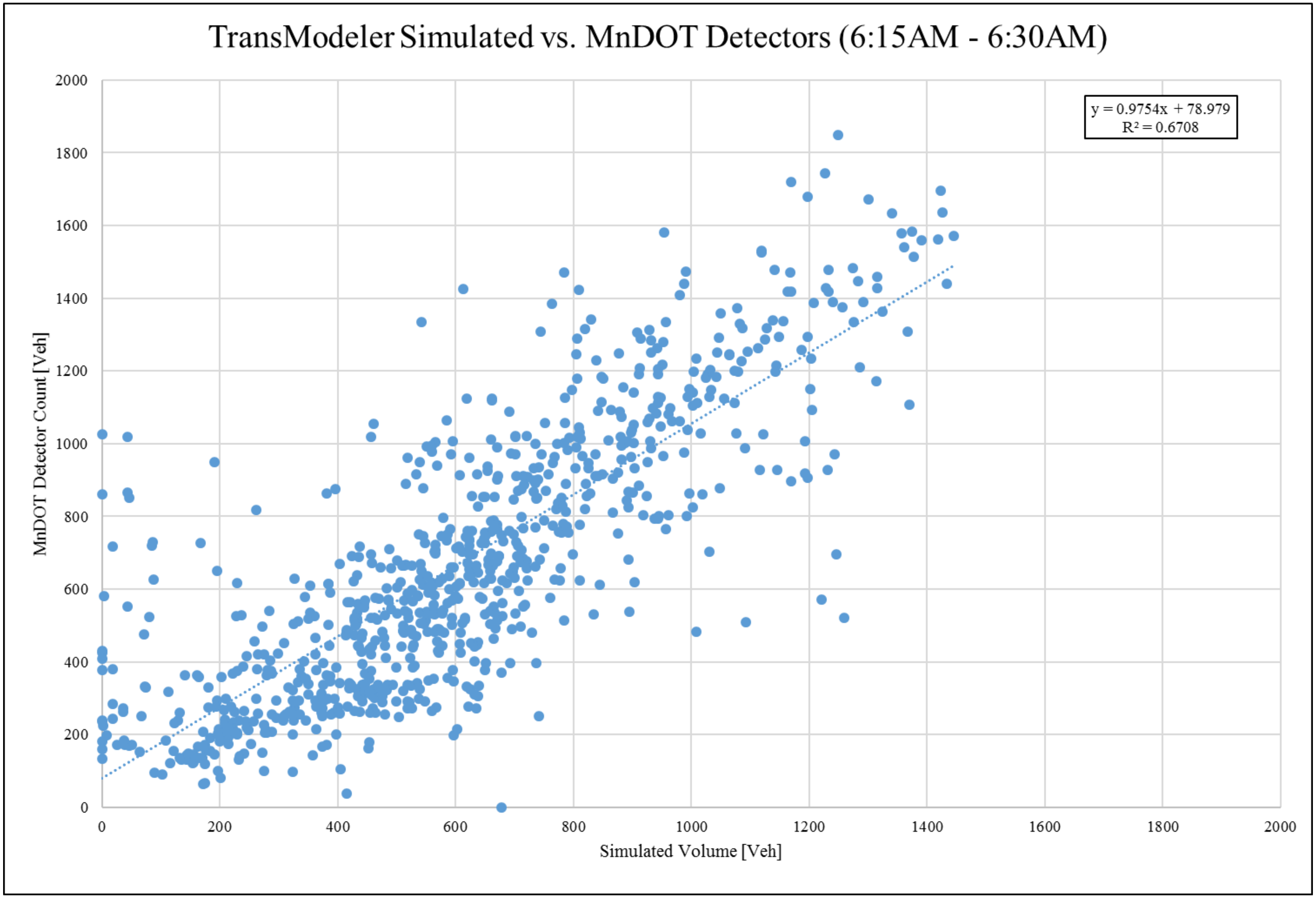
JACKSONVILLE DTA MODEL	
ROAD LENGTH	5,440 miles
LANE LENGTH	11,870 miles
ZONES (TAZ)	1,862 zones in the trip-based model (Trucks), and 492,684 parcels (Cars)
NUMBER OF SECTIONS	42,000 links and 31,000 centroid connectors
NUMBER OF INTERSECTIONS	NA There are 34,320 nodes, 1,300 of which are signalized intersections, and 14,508 of which are boundary nodes
SIMULATION DURATION	AM Peak 4 hr (5:00 – 9:00 AM) MD Peak 6.5 hr (9:00 AM – 3:30 PM) PM Peak 3 hr (3:30 – 6:30 PM).
TOTAL DEMAND	AM Peak ~ 738,000 MD Peak ~ 1,450,000 PM Peak ~ 866,000

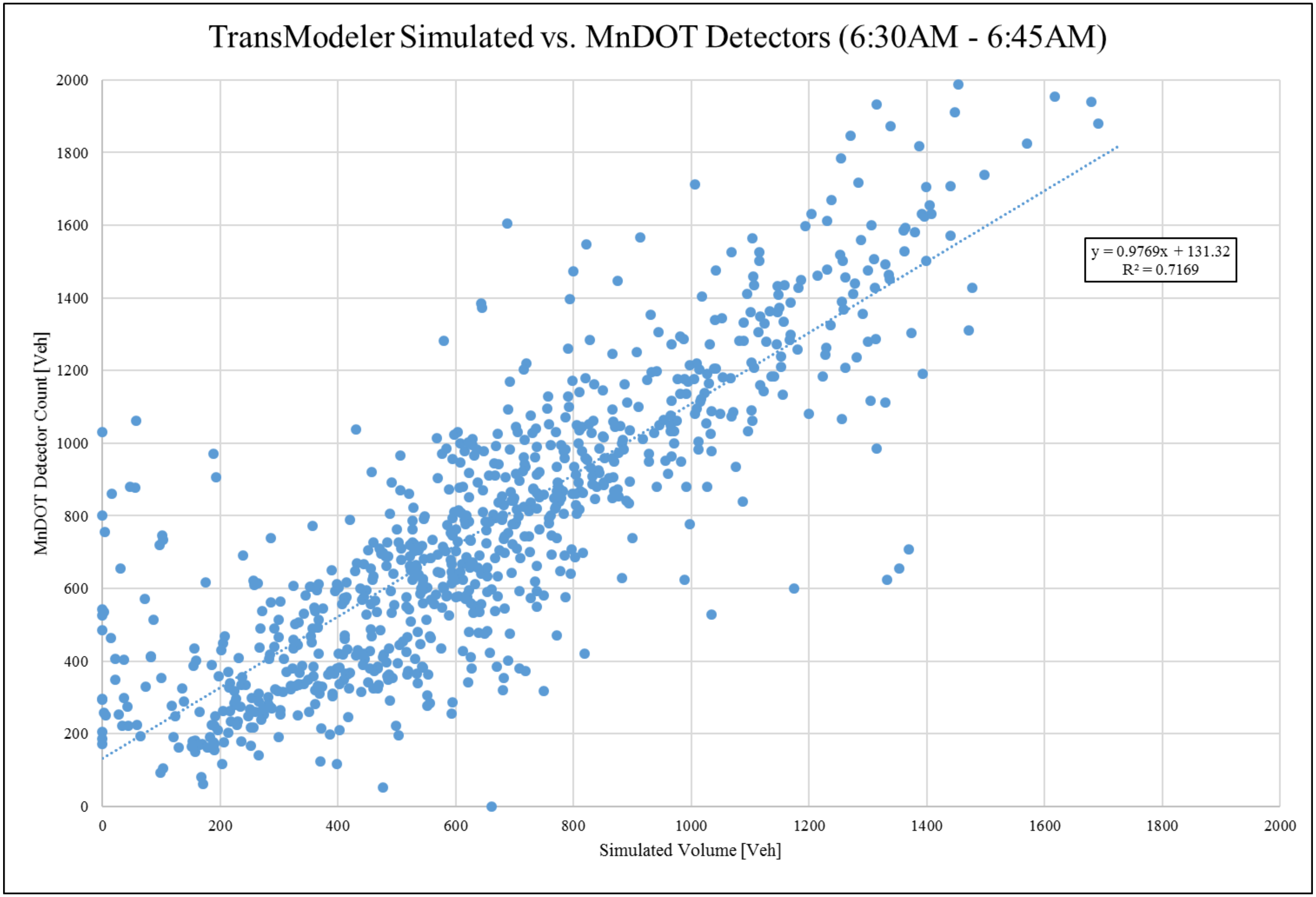
TIME TO SIMULATE	NA
VALIDATION CRITERION	%RMSE at 555 count locations
CLIENT	North Florida Transportation Planning Organization (NFTPO) and HNTB
DATE OF COMPLETION	2010
COST	NA[38]

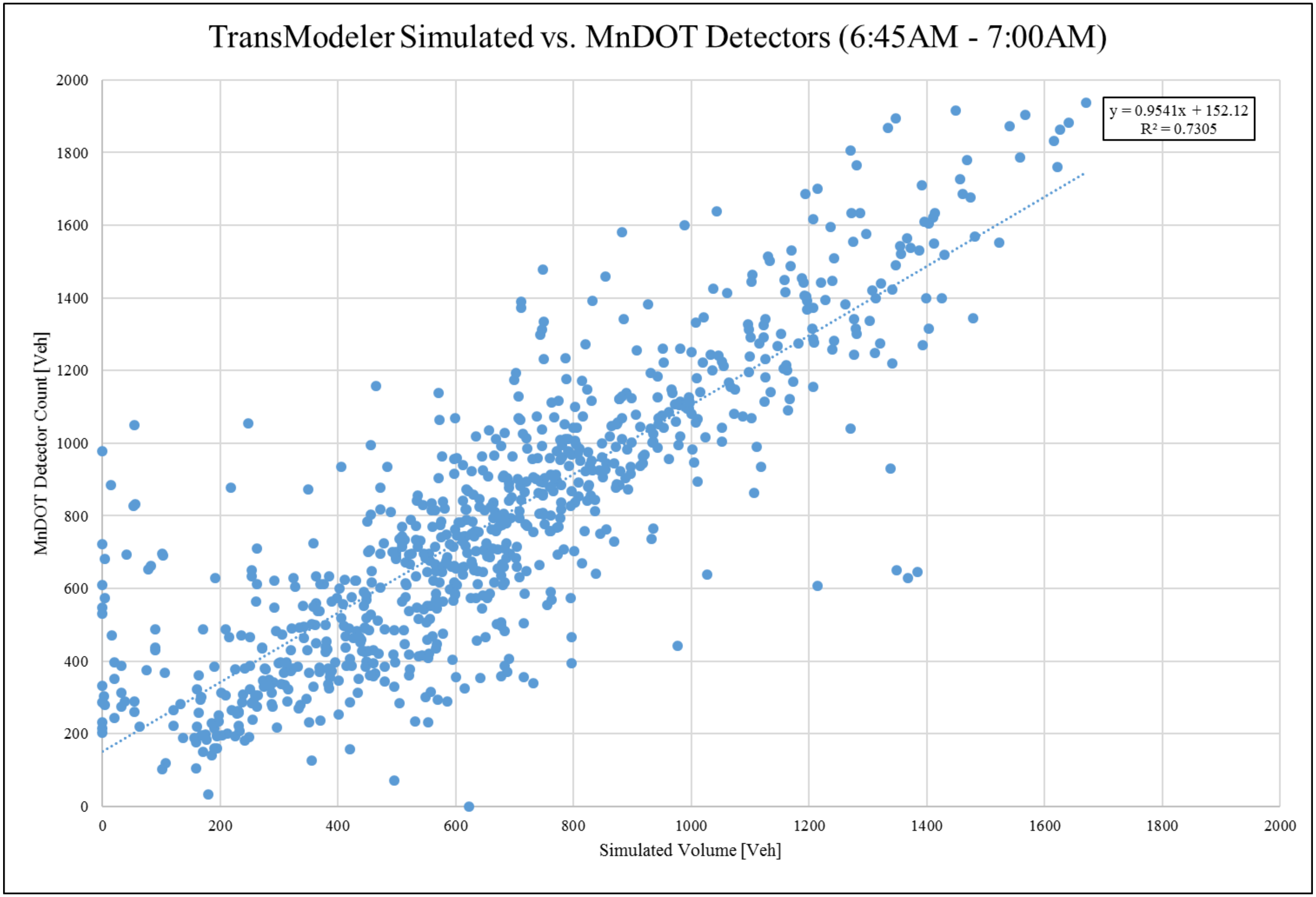
[38] Attempts were made to acquire the cost (both in time and money) but were not able to be acquired for this paper due to the reluctance of private companies and DOT's to disclose or inability to locate such statistics.

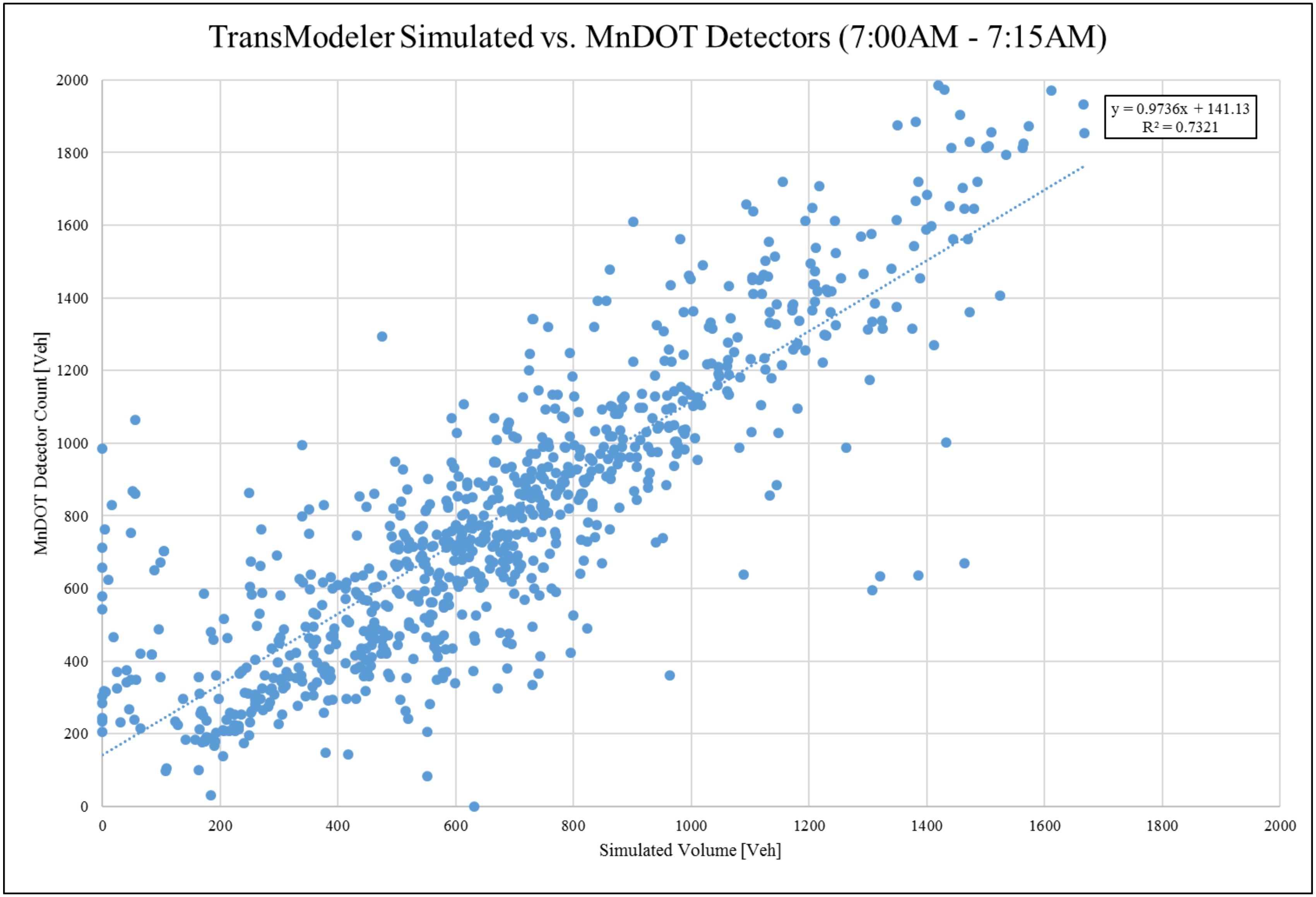
APPENDIX B - UNCALIBRATED TRANSMODELER RESULTS

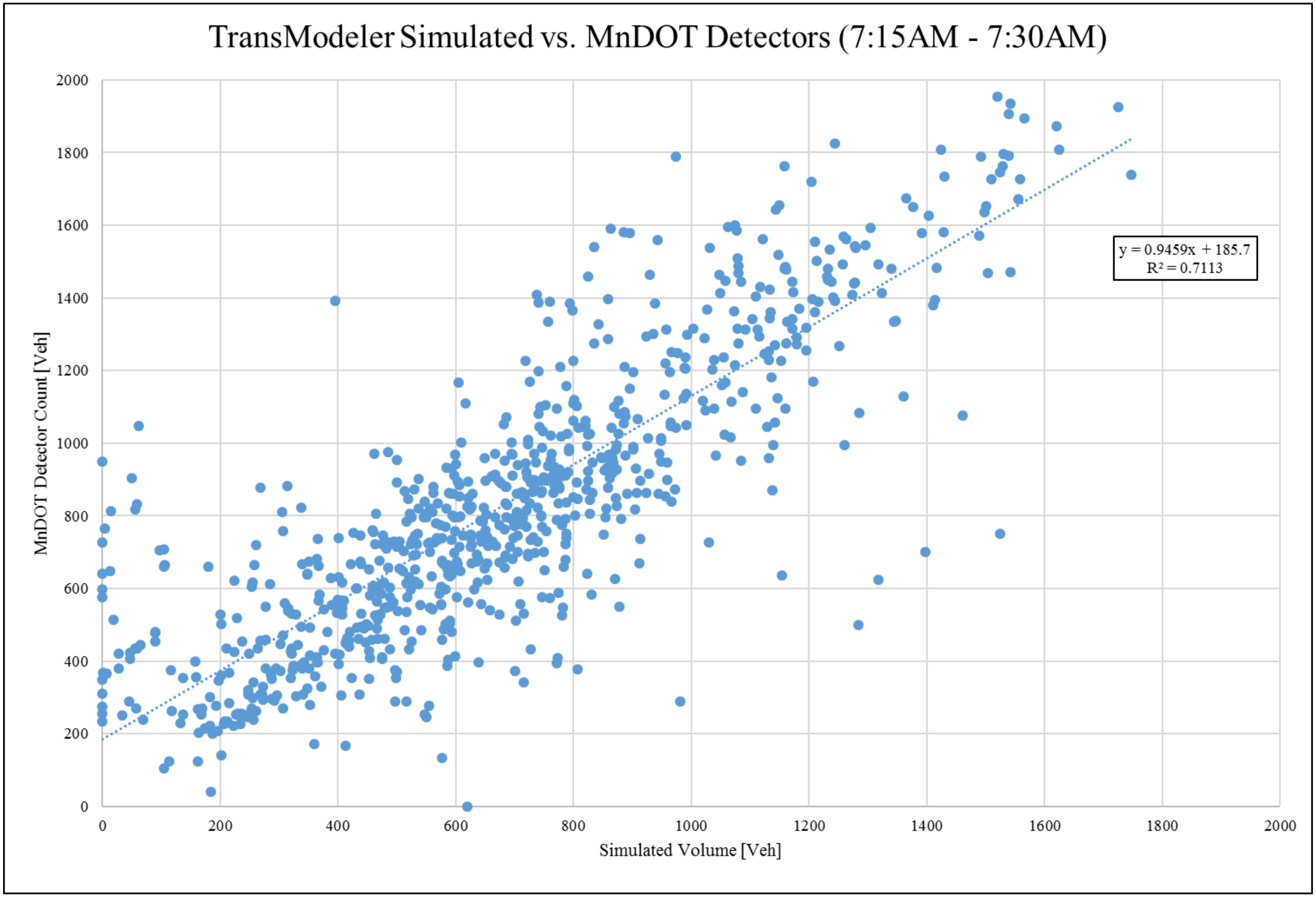

TransModeler Simulated vs. MnDOT Detectors (6:00AM - 6:15AM)
y = 0.6957x + 109.14
R² = 0.5229
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000

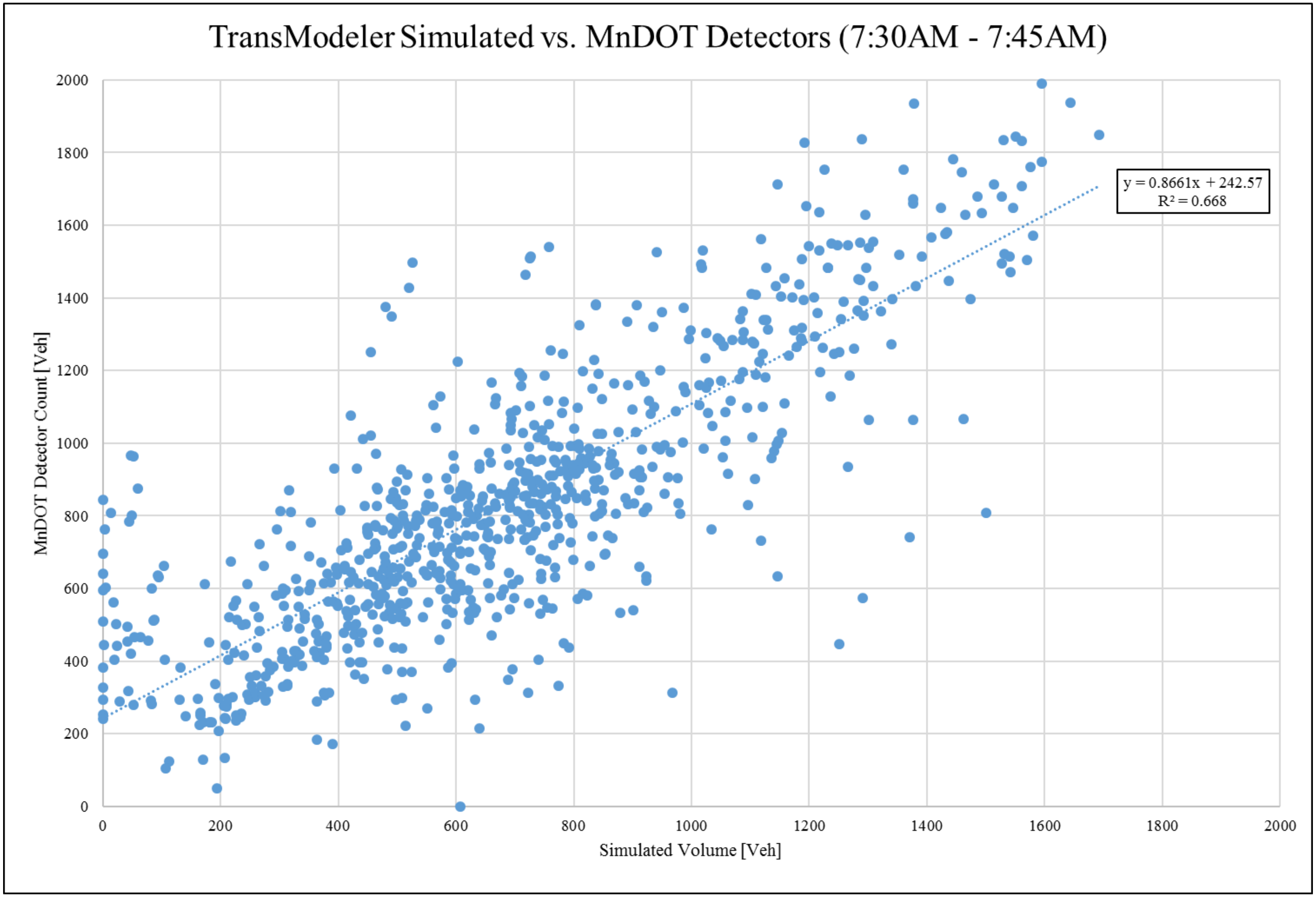

TransModeler Simulated vs. MnDOT Detectors (6:15AM - 6:30AM)
y = 0.9754x + 78.979
R² = 0.6708
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000

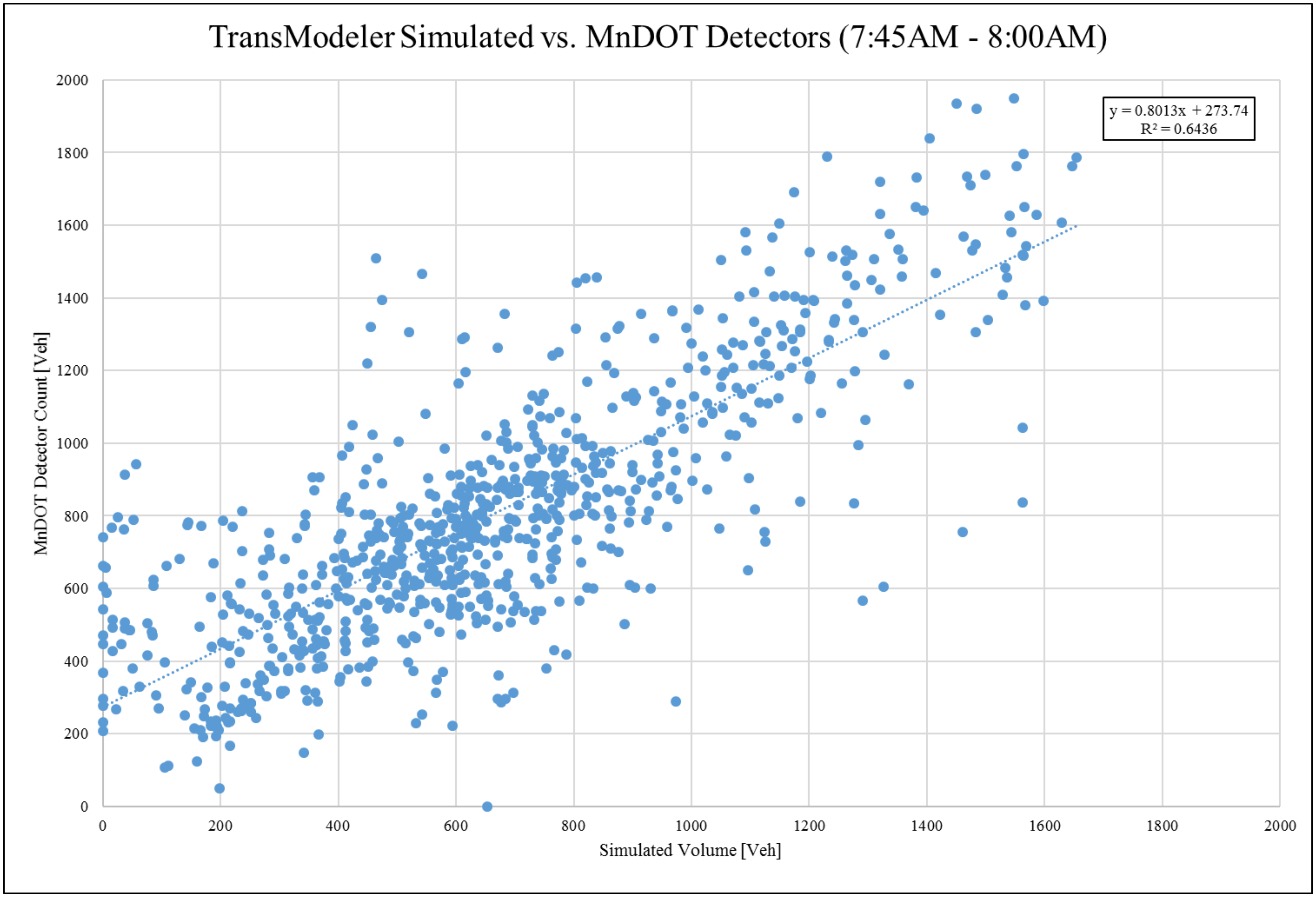

TransModeler Simulated vs. MnDOT Detectors (6:30AM - 6:45AM)
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000
y = 0.9769x + 131.32
R² = 0.7169

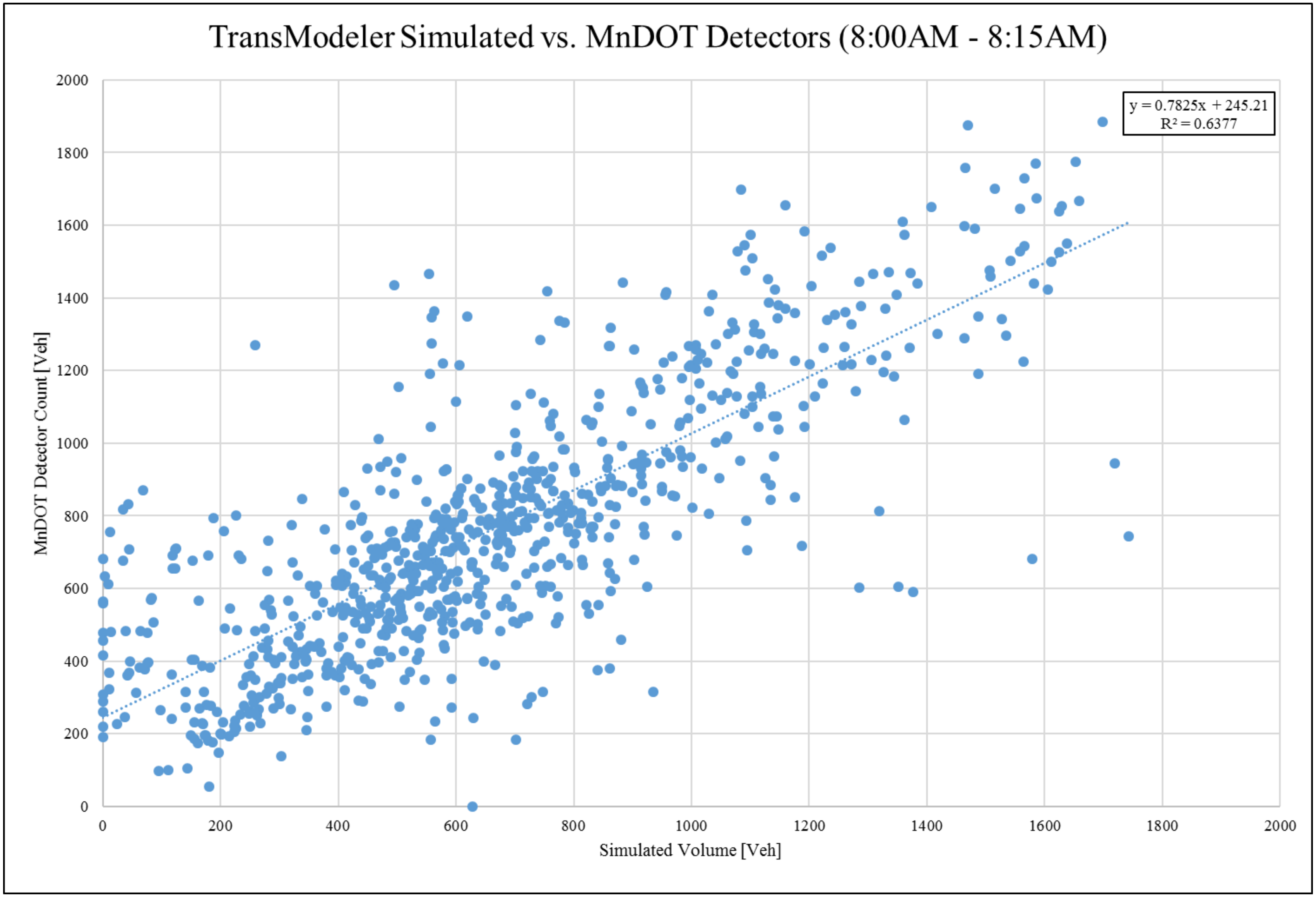

TransModeler Simulated vs. MnDOT Detectors (6:45AM - 7:00AM)
y = 0.9541x + 152.12
R² = 0.7305
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000

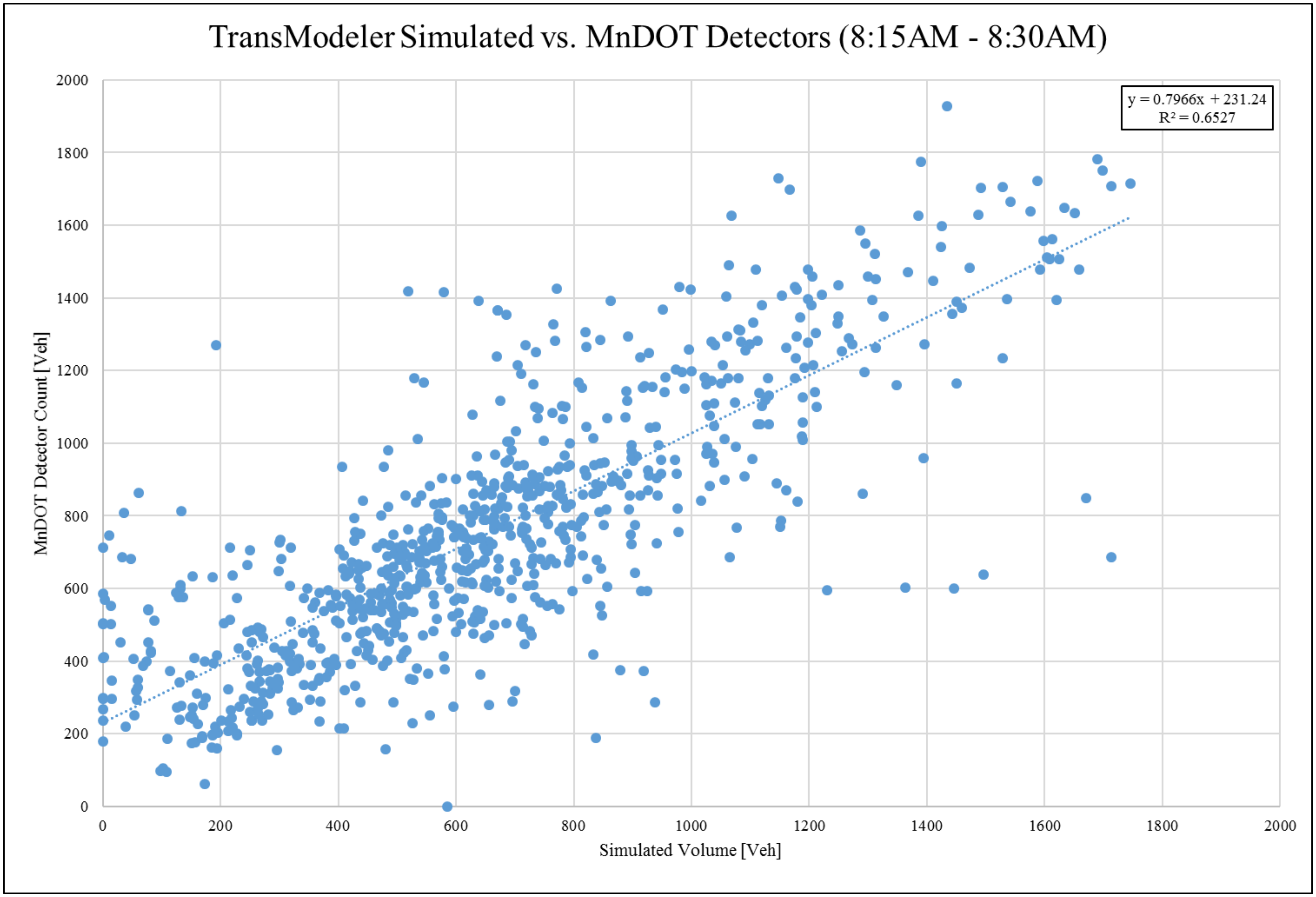

TransModeler Simulated vs. MnDOT Detectors (7:00AM - 7:15AM)
y = 0.9736x + 141.13
R² = 0.7321
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000

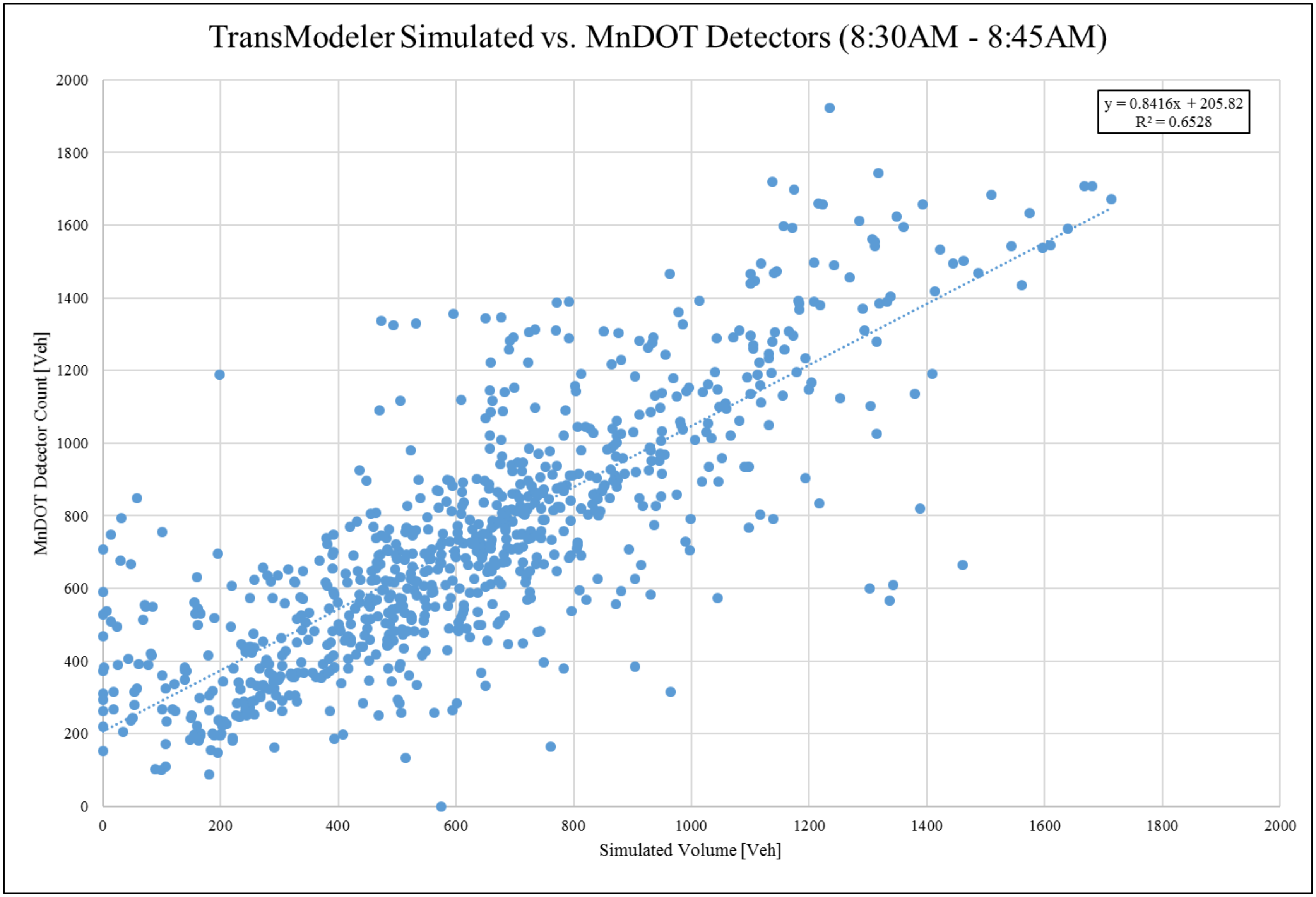

TransModeler Simulated vs. MnDOT Detectors (7:15AM - 7:30AM)
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000
y = 0.9459x + 185.7
R² = 0.7113

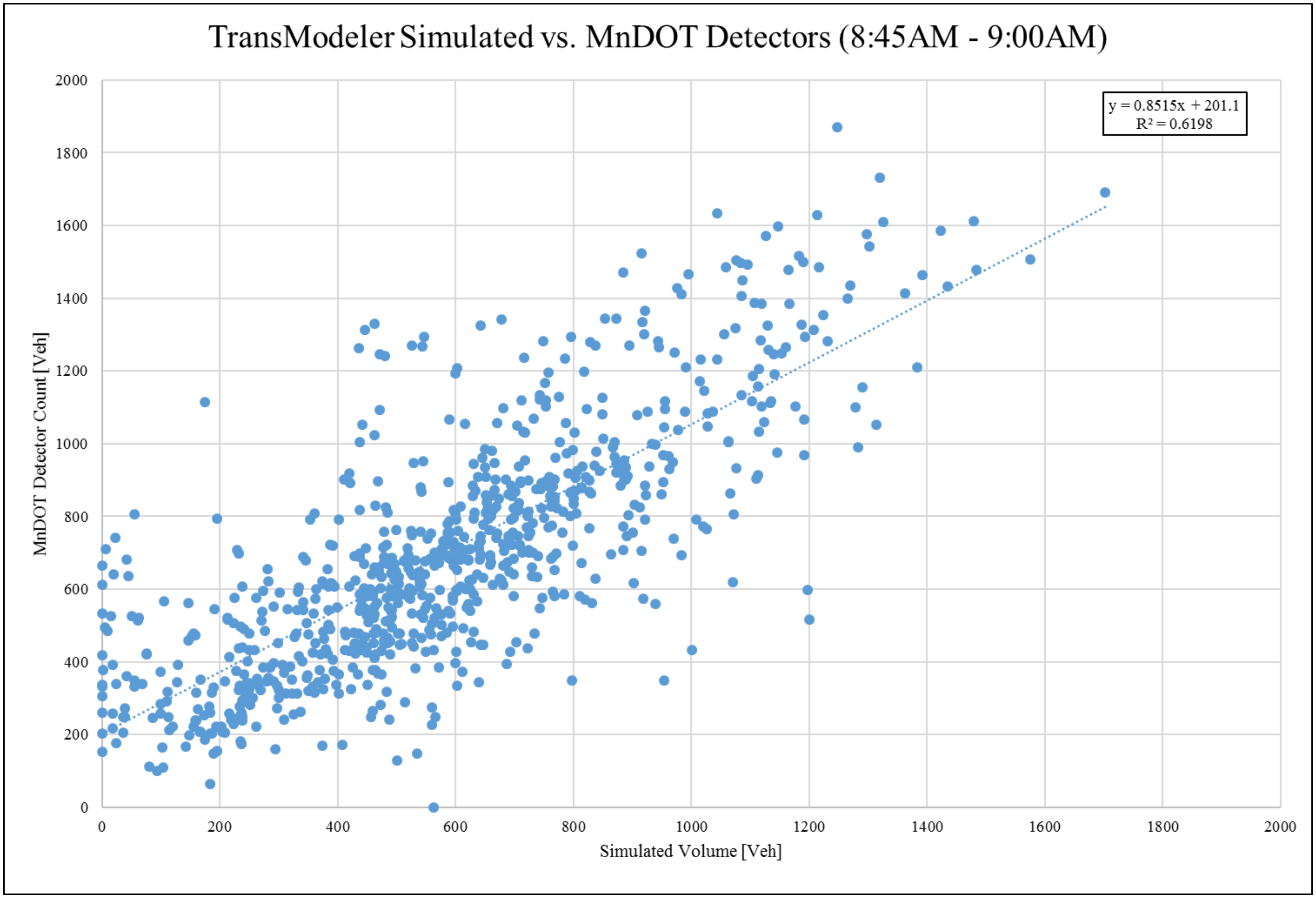

TransModeler Simulated vs. MnDOT Detectors (7:30AM - 7:45AM)
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
y = 0.8661x + 242.57
R² = 0.668
0
200
400
600
800
1000
1200
1400
1600
1800
2000


TransModeler Simulated vs. MnDOT Detectors (7:45AM - 8:00AM)
y = 0.8013x + 273.74
R² = 0.6436
MnDOT Detector Count [Veh]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Simulated Volume [Veh]


TransModeler Simulated vs. MnDOT Detectors (8:00AM - 8:15AM)
y = 0.7825x + 245.21
R² = 0.6377
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000


TransModeler Simulated vs. MnDOT Detectors (8:15AM - 8:30AM)
y = 0.7966x + 231.24
R² = 0.6527
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000


TransModeler Simulated vs. MnDOT Detectors (8:30AM - 8:45AM)
y = 0.8416x + 205.82
R² = 0.6528
MnDOT Detector Count [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000


TransModeler Simulated vs. MnDOT Detectors (8:45AM - 9:00AM)
y = 0.8515x + 201.1
R² = 0.6198
MnDOT Detector Count [Veh]
Simulated Volume [Veh]
0
200
400
600
800
1000
1200
1400
1600
1800
2000