Support for the Implementation of TRANSIMS in Portland, Oregon

Program Support for Environment and Planning
DTFH61-98-D-00108

Draft Technical Report

Using Traditional Model Data as Input to TRANSIMS Microsimulation

Prepared for

U.S. Department of Transportation
Federal Highway Administration

Prepared by

AECOM

July 11, 2006

Table of Contents

List of Tables

List of Figures

Preface

The Transportation Analysis and Simulation System, or TRANSIMS, is an integrated system of travel forecasting tools designed to give transportation planners accurate and complete information on traffic impacts, congestion, and pollution. It was developed by the Los Alamos National Laboratory (LANL) as part of the Travel Model Improvement Program (TMIP) sponsored by the U.S. Department of Transportation (USDOT), the Environmental Protection Agency (EPA), and the Department of Energy (DOE). TMIP was created to increase the ability of existing travel forecasting procedures to respond to emerging policy and technology issues. Moreover, TMIP includes long term efforts to redesign the travel forecasting process to reflect changes in behavior, respond to greater information needs placed on the forecasting process, and take advantage of changes in data collection technology. TRANSIMS is part of this long-term effort to redesign the modeling process from the ground up.

The goal of the TRANSIMS program is to develop technologies that offer transportation planning agencies increased policy sensitivity, more detailed vehicle-emission estimates, and improved analysis and visualization capabilities. The philosophy underlying TRANSIMS is that effective analysis of transportation systems requires detailed temporal and spatial simulation of travel conditions. The TRANSIMS Microsimulator tracks the activity patterns of individuals, households, and vehicles on a second by second basis in order to create a realistic representation of traffic dynamics and travel behavior. Additional information about TRANSIMS, technical documentation, and software downloads can be found on the TMIP Web site at http://tmip.fhwa.dot.gov/transims.

In 2002, the Federal Highway Administration (FHWA) sponsored a project entitled "Support for the Implementation of TRANSIMS in Portland, Oregon". The project was designed as a full implementation of the TRANSIMS software for regional modeling within the Portland metropolitan area. It followed efforts by LANL to use the Portland region as a proof of concept test case for a large scale implementation of activity-based microsimulation techniques. As part of the proof of concept, LANL developed, with the assistance of Portland METRO, a regional network that contained all of the local streets and transit routes within the greater Portland metropolitan area. This "All-Streets" network was used for the initial model development and software evaluation efforts.

In addition to the "All-Streets" implementation, FHWA pursued a parallel track designed to evaluate the feasibility of using the networks and trip tables from existing regional planning models as input to a TRANSIMS microsimulation. By comparing the results of the "Track 1" effort to the more detailed "All-Streets" implementation, FHWA could evaluate the relative costs and benefits of pursuing one approach over the other. If the "Track 1" approach generated reasonable results, it would be a quick and cost-effective way for regional planning agencies to investigate the potential of regional simulation as an extension or enhancement to current planning practices.

The Portland study also included the implementation of a complete activity-based regional model using the TRANSIMS Population Synthesizer, Activity Generator, Router and Microsimulator. One of the objectives of this effort was to utilize existing trip-based models to the full extent possible. If trip-based models could be re-calibrated within a TRANSIMS framework, an effective transition strategy from current practice to advanced activity-based analysis could be demonstrated. The so called "GEN2" model incorporates several components of the existing Portland regional model into a TRANSIMS model that simulates individual and vehicle movements on a second by second basis.

A number of reports were generated by the Portland implementation project. These reports are available on the TMIP Web site http://tmip.fhwa.dot.gov/transims. These include:

1. Using Traditional Model Data as Input to TRANSIMS Microsimulation, January 2006.

2. Generating TRANSIMS Activities from Trip-based Location and Mode Choice Models, May 2006.

Presentations and papers were submitted to the Transportation Research Board (TRB) Annual Meetings and the TRB's bi-annual Transportation Planning Applications Conferences. These include:

1. "Using Traditional Model Data for Microsimulation and Emission Estimates", Transportation Planning Applications Conference, Baton Rouge, Louisiana, April 8, 2003.

2. "Using Microsimulation to Estimate Emissions – a Comparison of TRANSIMS and MOBILE6", Transportation Planning Applications Conference, Baton Rouge, Louisiana, April 8, 2003.

3. "TRANSIMS GEN2 Model Specifications for the Portland Test Case", Transportation Planning Applications Conference, Baton Rouge, Louisiana, April 8, 2003.

4. "Micro Simulation Application – TRANSIMS in Portland Oregon", TRB 83rd Annual Meeting, Washington, D.C., January 11, 2004.

5. "The Impact of Regional Simulation on Emission Estimates", TRB 84th Annual Meeting, Washington, D.C., January 11, 2005.

6. "Using Traditional Model Data as Input to TRANSIMS", Transportation Planning Applications Conference, Portland, Oregon, April 25, 2005.

7. "Practical Tour-based Models for Microsimulation of the Portland Region", Transportation Planning Applications Conference, Portland, Oregon, April 25, 2005.

8. "Activity Modeling in Portland", TRB 85th Annual Meeting, Washington, D.C., January 23, 2006.

Chapter 1 Introduction

This paper was developed as part of a Federal Highway Administration (FHWA) project entitled “Support for the Implementation of TRANSIMS in Portland, Oregon”. The project was designed as a full implementation of the TRANSIMS software for regional modeling within the Portland metropolitan area. The project included a number of research tracks. “Track 1” examined the feasibility of using the networks and trip tables from existing regional planning models as input to a TRANSIMS microsimulation. If the “Track 1” approach generated reasonable results, it would be a quick and cost-effective way for regional planning agencies to investigate the potential of regional simulation as an extension or enhancement to current planning practices.

This paper documents the “Track 1” effort. This effort used Portland METRO’s existing regional model networks and trip tables to generate the data needed by TRANSIMS to execute the Router and Microsimulator modules. The Router and Microsimulator are the highway and transit assignment components of the TRANSIMS modeling approach. This “Track 1” approach does not replace the Trip Generation, Trip Distribution, or Mode Choice components of the regional modeling process. Integration of these components into a complete TRANSIMS implementation is addressed in other documents.

1.1 Network Conversion

The “Track 1” approach used traditional model data from Portland METRO’s trip-based model to generate the inputs needed by the TRANSIMS simulation process. Highway and transit networks were converted to TRANSIMS network files. Synthetic generation procedures were used to create pocket lanes, lane connections, signal timing plans, and other inputs required by TRANSIMS. Chapter 2 details the network conversion process that was used to generate synthetic TRANSIMS network files from METRO’s regional network data.

METRO’s highway network included 11,000 two-way links and 1,260 zones covering the Portland metropolitan area and Clark County (Vancouver) in the state of Washington. This network resulted in 8,300 links, 8,000 lane miles, 1,730 traffic signals, and 20,300 activity locations (i.e., zones) within the TRANSIMS environment. This compares to 120,000 links, 30,000 lane miles, 1,750 traffic signals, and 240,000 activity locations in the “All-Streets” network initially developed for the case study. In other words, the MPO-level network includes 26 percent of the lane miles and virtually all of the signalized intersections, but only seven percent of the links within the Portland region. Since the vast majority of

vehicle miles of travel are made on primary facilities, the MPO-level network serves the primary objectives of regional transportation planning.

1.2 Trip Table Conversion

The TRANSIMS architecture is based on the time, duration, and location of each person’s activities throughout the course of the day. Persons are associated with households in order to consider the interrelationships of activities at the household level. Household composition is also a useful predictor of the types of activities a given person is likely to make. Given this focus, the TRANSIMS software includes a Population Synthesizer to generate households of various compositions based on Census information and growth forecasts. The Activity Generator is then used to assign activity patterns to these households and locate the activities within the region. In this construct, travel becomes the mechanism by which a person moves from one activity location to another.

Since all travel within a region is not made by local household members, TRANSIMS also includes the concept of itinerate trips to capture the impacts of these trips on network performance. Itinerant trips include trips made by people from outside of the region, trips that travel through the region, and trips made by commercial vehicles within the region. The modeler includes these trips in the simulation through zone-to-zone trip tables and diurnal distribution curves. The software allocates the zone-based trips to activity locations within the zone and selects the time of day for the trip based on the diurnal distribution. A single person household and a vehicle are created at the origin of the trip and an activity for this person is defined at the trip destination. This means that itinerant trips can be modeled in exactly the same way as household-related trips.

The approach described in this document takes the concept of itinerant trips one step further. Rather than use the Population Synthesizer and Activity Generator to create households and activity patterns, this approach uses the itinerant trip method to create activities for the internal person trips as well. This allows the TRANSIMS Router and Microsimulator process to be applied using standard trip tables produced by the regional travel demand forecasting model.

1.3 Assignment Techniques

TRANSIMS was conceived from the premise that travel demand will emerge as a result of a series of feedback loops. In other words, as people learn about travel opportunities and system performance, they adjust their behavior to satisfy their needs (i.e., activities). From an application perspective, feedback is an iterative process of identifying the best way for each traveler to complete their daily activities. The link travel times generated by the Microsimulator are used during the feedback process to adjust travel paths. The current TRANSIMS algorithm and application methodology does not, however, attempt to solve this problem for millions of travelers simultaneously. It approaches the problem as a series of incremental adjustments made by a subset of travelers that eventually migrate toward acceptable travel conditions for all travelers. This differs from traditional models where feedback is always applied to all travelers simultaneously. In other words, we re-run the complete model using the latest speeds. This approach tends to create oscillation effects that make model convergence more difficult. This study implements a number of different types of feedback loops to address specific problems or opportunities for a particular selected subset of travelers who find themselves is specific situations (e.g., their current route is highly congested, they can't get to work on time, transit is not available at the time they need to travel, they have a hard time making a left turn at a given intersection, etc.).

The other important concept to understand about TRANSIMS is computational partitioning. The TRANSIMS software is specifically designed to perform microscopic simulations of daily travel in large regions with detailed networks. The Microsimulator is computationally intensive and takes a considerable time to run when used for regional simulation. In order to make the process efficient, the TRANSIMS software splits the calculations into independent tasks that can be executed on different CPUs in a computer cluster. The results of these computations are then merged together to reflect the performance of the entire system. Partitioning makes the process feasible and time efficient, but it also creates implementation issues that are not encountered during a single CPU application.

In its most basic form, a TRANSIMS assignment is similar to traditional assignment techniques. In the TRANSIMS application, minimum time or impedance paths (also referred to as travel plans) are constructed for each trip based on the latest link travel times. The trips are added to the links on the path and the resulting traffic volumes are used to re-estimate travel times. The process continues until some stopping criteria are reached. In traditional models these criteria might be the number of iterations and/or a convergence statistic from a system or user equilibrium calculation. In traditional software packages, this whole process takes place within the assignment module. In TRANSIMS this process is typically identified as a Router-Microsimulator feedback process. The process involves a number of programs that can be combined in a variety of ways to model the transportation system and demand for travel. The primary difference between the TRANSIMS assignment and the traditional models is that the TRANSIMS application considers time dependent travel times while determining the minimum time or impedance paths. In the TRANSIMS application, the travel times are typically summarized for 15 minute time intervals. The second major difference between the TRANSIMS application and traditional models is that the TRANSIMS application attempts to minimize the path for each and every trip where as the traditional models tries to achieve system wide equilibrium.

1.4 Model Validation and Sensitivity Tests

In many respects, this research is intended to investigate the methods and procedures required to use the TRANSIMS Router and Microsimulator to build paths and simulate traffic flow through a network. The objective was to test a variety of methods that could efficiently stabilize the traffic volumes and speeds while at the same time minimize the travel times for each trip. Review of the simulation results using the Output Visualizer showed that the Microsimulator generated logical levels of congestion by time of day.

Generating reasonable looking results, however, is only the first step in developing a modeling process that can be used to evaluate network alternatives and future travel conditions. The modeler must demonstrate that the process replicates observed travel conditions for the base year and behaves logically when the underlying data are changed. This is the process of model validation and sensitivity testing. This paper documents the initial efforts to convert a research tool into a practical applications model.

The validation effort included a number of traditional and non-traditional performance measures. Traditional measures included comparisons of assigned volumes to observed traffic counts on screenlines, facility types, area types and volume group. Some non-traditional measures included distributions of traffic and speeds by time of day. It was also necessary to calibrate and validate the simulation model itself in order to accurately replicate the traffic throughput observed in Portland.

Once the TRANSIMS model replicated the observed traffic counts, diurnal distributions, and speeds reasonably well, a number of tests were performed to document the sensitivity of the network performance and the user-equilibrium condition to changes in supply and demand. The sensitivity tests were designed to highlight the behavior of the model in future year forecasting and alternatives analysis. The tests were limited to relatively simple changes to the number of trips or the base year network. More extensive network and travel changes would be necessary to conclude that the process generates robust results over the full range of scenarios typically included in long-range transportation planning. These studies are likely to be part of FHWA's continuing research efforts.

Chapter 2 Network Preparation and Editing

The process of converting METRO's regional network to TRANSIMS is summarized in Figure 1. Highway and transit network files are first exported from the EMME/2 databank to standard ASCII record formats. The utility program called Emme2Net is used to convert the network records to a generic TRANSIMS link-node format. This process strips out zone centroids and zone connectors, combines one-way links into two way links, and calculates the number of lanes, speeds, capacities, and use restrictions for each link. The generic link-node data are read by the TransimsNet program to generate the additional network files needed by TRANSIMS. These files include the activity locations, parking lots, and process links required for trip making and the pocket lanes and lane connectivity needed to move vehicles from one link to another. The TransimsNet program also creates signal and sign warrant files for each intersection. These files are subsequently used in the IntControl and Progression programs to generate all of the intersection control files needed by TRANSIMS.

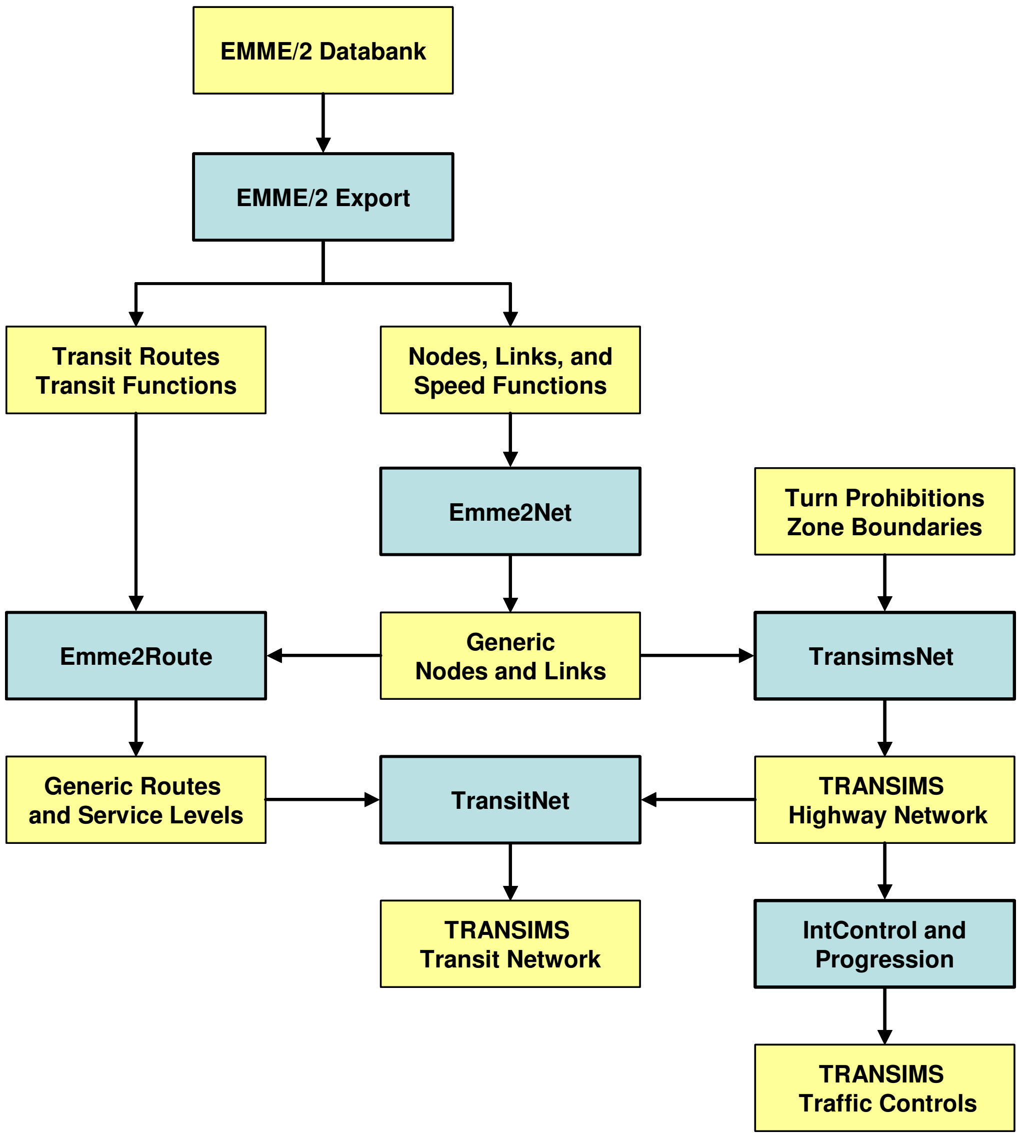

EMME/2 Databank
EMME/2 Export
Transit Routes Transit Functions
Nodes, Links, and Speed Functions
Emme2Net
Turn Prohibitions Zone Boundaries
Emme2Route
Generic Nodes and Links
TransimsNet
Generic Routes and Service Levels
TransitNet
TRANSIMS Highway Network
TRANSIMS Transit Network
IntControl and Progression
TRANSIMS Traffic Controls

FIGURE 1: Network Conversion Process

The generic link-node file is also used by the Emme2Route program to convert the transit route records to a generic TRANSIMS route format. The network files generated by TransimsNet are combined with the generic transit route file within the TransitNet program to generate the TRANSIMS transit network files. These files include the transit stops, routes, schedules, and driver plans.

Throughout the conversion process, the user is provided with opportunities to review and edit the intermediate files before continuing. The files generated at each step in the process can be converted to ArcView shape files using the ArcNet program to facilitate review and editing. Alternatively, the TRANSIMS files can be reviewed and edited using spreadsheet software or standard text editors. A number of recommended checks and refinements are highlighted in the discussion that follows. After the initial checks are made, the user will typically read the files into the TRANSIMS Network Editor and apply the ValidateNetwork program to verify that the changes have been implemented correctly.

For all practical purposes, the user review is the only time consuming part of the process. The software tools only take a few minutes to generate the full set of TRANSIMS files. If the synthetic results are not acceptable, there are several parameters at each step in the process the user can adjust to create more realistic results. A moderate amount of manual corrections are recommended as part of any network conversion. These corrections will typically focus on major freeway interchanges and complicated arterial intersections. About two weeks worth of user review is generally adequate for most networks. A detailed review of each intersection can typically wait until problems are identified by the Microsimulator.

The remainder of this chapter describes each step in the conversion process in more detail and discusses the issues that were encountered in Portland and how these issues were addressed.

2.1 Generic Network Preparation

The conversion process is implemented in two primary steps. The first step converts the highway and transit data from a specific travel demand software package into a generic TRANSIMS network format. The second step converts the generic link-node-route information into the detailed file structure needed by TRANSIMS. This section discusses the process of converting the Portland network to the generic TRANSIMS network format. (Note: utilities have been developed to convert TP-Plus and ArcView network data into this same format. Users may prefer to write their own software to manipulate the network data).

2.1.1 EMME/2 Highway Network

The Emme2Net program is used to create generic TRANSIMS link and node files from EMME/2 network card images. To begin the conversion process, the user exports or "punches out" the link, node, and function information from the EMME/2 databank. The Emme2Net program reads files in the standard EMME/2 "card image" ASCII format. This format lists the nodes at the beginning of the file followed by directional links. User field #3 should be used to export the total hourly capacity for each link record. If the volume-delay functions are exported, the function records are used to lookup the free flow speed for each link.

The primary challenge in converting an EMME/2 network to TRANSIMS is converting the total hourly capacity and volume-delay function numbers used by EMME/2 into the number of lanes, speed limit, and functional classification required by TRANSIMS. In the Portland study, the conversion was complicated by the evolutionary process by which the METRO network was created. The original EMME/2 network was developed for Portland using relatively simple coding techniques. As the EMME/2 software and computer technology evolved, the network was enlarged and additional detail was included (e.g., directional freeways and ramps). The City of Portland also used different coding conventions for streets in residential areas. This created discrepancies in the definitions of major and minor arterials for streets inside and outside the City of Portland. This was further complicated when the network developed for Clark County (Vancouver, Washington) was merged with the Oregon network to create a single regional network. As you might expect, each network used slightly different definitions of facility types, lanes, and speeds.

The METRO network also included a number of special coding rules that work well within EMME/2, but cause difficulties for TRANSIMS. For example, METRO coded a center lane that can be used for left turns in either direction as an additional half lane in both directions. This provided EMME/2 with the additional capacity it needed to account for the center turn lane in the highway assignment process. Since TRANSIMS models both regular and pocket lanes, a realistic simulation of center turn lanes requires elaborate coding of alternating pocket lanes. The synthetic process needed to address these conditions with more straightforward techniques.

To address the network anomalies and provide the user with more control over the conversion process, lane capacity rules were added to the conversion software. A lane capacity file is used to constrain the relationships between functional class, speed, and lane capacity. It also defines the types of adjustments that are made when the data are outside of the logical constraints. The lane capacity file is a comma delimited data file typically generated by standard spreadsheet software. It contains the following information:

1. Input functional class
2. Free flow speed (mph)
3. Minimum hourly capacity
4. Maximum hourly capacity
5. Default hourly capacity
6. Minimum output functional class
7. Maximum output functional class
8. Default output functional class

The user can provide up to nine input functional classes and up to fourteen 5 mph-speed increments. If a category is not provided, the program uses default values by functional class to populate missing fields. A multi-step process is used to convert the data on the input network file into logical combinations of functional class, speed, and number of lanes needed by TRANSIMS. The data provided in the lane capacity file defines the limits within which adjustments are made. The process starts by using the volume delay function number and the functional class found on the input network file to select a constraint record from the lane capacity table. It then uses the capacity constraints to calculate the number of lanes from the total hourly capacity. If the number of lanes is illogical for the given functional class, functional class and capacity adjustments are applied until a logical combination of functional class, speed, and number of lanes is identified. The user can then look at the results within ArcGIS and make adjustments as necessary.

2.1.2 Generic Highway Network

Figure 2 shows how the Emme2Net program is used to create a generic network. The input highway network and function files are standard EMME/2 card image files. The lane capacity file is the spreadsheet file discussed above that facilitates adjustments to the functional class, number of lanes, and speed limit relationships. The user can also provide parameters to implement a coordinate conversion, map the functional class codes to TRANSIMS functional class codes, and define external station zone numbers. Since the TRANSIMS coordinates need to be in UTM meters, the Emme2Net program includes parameters to convert Latitude/Longitude or Stateplane coordinates to UTM. External stations also require special treatment within TRANSIMS. This processing is facilitated by keeping the zone connectors attached to external station zones in the output network file. These links are used by the TransimsNet program to create the appropriate external parking lot links.

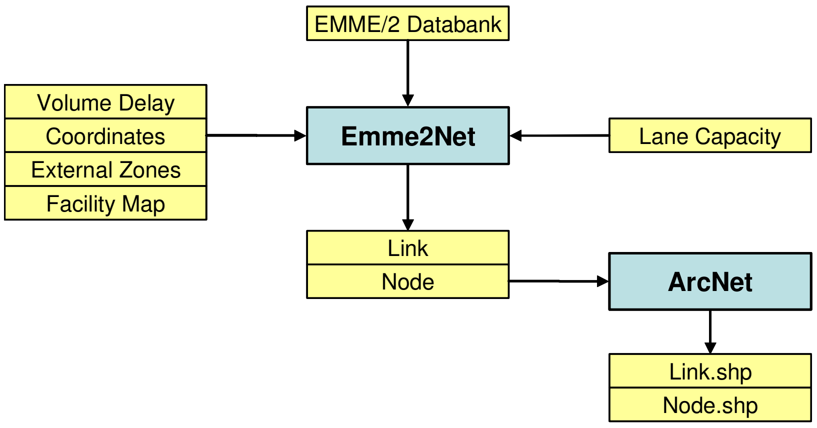


FIGURE 2: Generic Highway Network

The program outputs a list of the links where the functional class, number of lanes, and speed limit relationships were adjusted. It also generates a link and node file in TRANSIMS format (i.e., a tab-delimited ASCII file) with many but not all of the data fields and coding conventions used in TRANSIMS. The node file includes three fields:

1. ID
2. EASTING
3. NORTHING

The generic link file includes the critical fields from the TRANSIMS link file. These fields include:

1. NODEA
2. NODEB
3. PERMLANESA
4. PERMLANESB
5. LENGTH
6. CAPACITYA
7. CAPACITYB
8. FREESPDA
9. FREESPDB
10. FUNCTCLASS
11. VEHICLE

As long as the field names are consistent, the fields do not need to be provided in any particular order. These are all of the data items needed to construct a synthetic TRANSIMS network. These files can also be converted to ArcView shape files using the ArcNet program. Displaying the network in ArcGIS is a useful way of reviewing the conversion results and making adjustments as necessary to more accurately represent the functional class and number of lanes assigned to a given link. It is fairly important to have these two data items accurate before proceeding to the TRANSIMS network generation.

2.1.3 EMME/2 Transit Network

The Emme2Route program generates the generic route header and route node files from the EMME/2 route card images. The generic route file stores information about the mode, stops, dwell times, travel times, and headways for different time periods for the route. Portland METRO coded transit routes using round trip node lists and a single headway value. PM peak period and off-peak routes were coded as separate networks. In order to make this information useful for TRANSIMS, a number of changes were made to the route coding.

The first change was to combine the PM peak and off-peak routes into a single route file. Since most routes had identical paths in the PM peak and off-peak networks, this was relatively straightforward. The second step was to split the routes into in-bound and out-bound orientations. In most cases, METRO included a layover field in the node list at the turn-around point. A text editor was used to split the route into two routes at the layover point in the route. The route name was modified to include an inbound / outbound code.

Combining the two networks and splitting the routes were needed to define the service characteristics by time of day. Since TRANSIMS simulates travel for the whole day, the transit service available at a given time of day is important for modeling transit trips. Portland METRO simulated the service by time of day using seven time periods:

1. 0:00-5:00
2. 5:00-7:00
3. 7:00-8:30
4. 8:30-16:00
5. 16:00-18:00
6. 18:00-21:30
7. 21:30-24:00

METRO added a comment record to each of the routes in the card image file. The record included a headway and offset value for each of the seven time periods. The PM peak and off-peak headway information was derived from the original network files. Headway data for other time periods was extracted from published Tri-Met schedules.

The offset information is used to define the start time for the first run within a given time period. In most cases, METRO used the random generation option to set the first run offset. This option sets the offset as a random percentage of the time period headway. For example, if a route is assigned a 30 minute headway in a given time period, the first run of that route will be randomly assigned between 0 and 29 minutes from the start of the time period. This distributes the start times for the routes and helps to avoid bunching the routes on common route segments.

There were routes, however, that required better coordination to simulate the relative service level of common segments. Several routes share a common segment for the majority of the route and only vary at one end or the other. In these cases, METRO wanted to control the headway offsets on the two routes in order to minimize the waiting time on the common segment. METRO coded one route with a zero offset and set the offset for the other route to one-half of the headway. For example, the "Division" route was coded with a "long" and "short" route configuration. Both routes had a 60 minute off-peak headway that was coordinated in the real world to provide 30 minute service for trips served by the "short" route. In this case the "long" route was coded with an offset of zero and the "short" route was assigned a 30 minute offset.

The total time required to convert the original network coding to this new format was about two weeks. Since the changes were made directly to the ASCII card images, METRO was able to re-load the card images into the EMME/2 software and use the EMME/2 network editor to review and edit much of the route information. This also made it easier to adjust the transit routes to compensate for issues found by the TRANSIMS simulation. Most of these involved end-of-route and turn-around issues that do not affect EMME/2, but needs to be corrected for TRANSIMS.

2.1.4 Generic Transit Network

Figure 3 shows how the Emme2Route program is used to create a generic transit network. The input route and travel time function files are standard EMME/2 card images. The route file was enhanced to include the additional headway and offset information discussed above. The function file defines how the transit travel times relate to the speed values included in the input link file. The input link file is the output file from the Emme2Net process. The user also specifies the number of time periods and the minimum dwell time per transit stop.

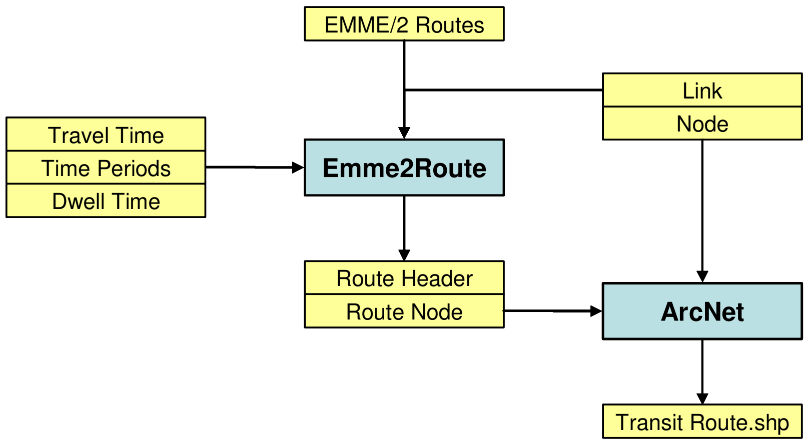


FIGURE 3: Generic Transit Network

The program outputs the generic route header and route node files. The route header file is a tab-delimited ASCII file with one record per route. Each record includes the following fields:

1. ROUTE
2. NAME
3. MODE

4. TTIME
5. HEADWAY_x
6. OFFSET_x

where the "_x" after each headway and offset field identifies the sequence number of time periods specified by the user. For the Portland study, seven time periods were defined (0:00-5:00, 5:00-7:00, 7:00-8:30, 8:30-16:00, 16:00-18:00, 18:00-21:30, and 21:30-24:00). A headway and offset pair is provided for each time period. A zero headway indicates that the route does not provide service during that time period.

The route node file is a tab-delimited ASCII file with a record for each network node included on each route. The node fields include:

1. ROUTE
2. NODE
3. DWELL
4. TTIME
5. SPEED

The nodes must be specified in order. Nodes that include dwell time are assumed to be stop nodes. If the dwell time is zero, the route will pass through the node, but not stop. These are all of the data items needed to construct a synthetic TRANSIMS transit network. The ArcNet program can convert these files to ArcView shape files for review and editing.

2.2 Synthetic Network Generation

The synthetic network generation was split into three major steps to provide flexibility and permit the review and editing of intermediate results. The first step converts the generic link and node data to a basic TRANSIMS highway network. This process does not generate a full set of intersection traffic controls, but a list of locations where signs and signals are warranted. The user typically reviews the signal and sign warrants before proceeding to the second step where the intersection traffic control files are generated. The third step is the optional conversion of the generic transit route data to a TRANSIMS transit network with the required access components.

2.2.1 TRANSIMS Highway Network

The TransimsNet program was used to generate a synthetic TRANSIMS highway network using the generic link and node files generated by the Emme2Net program. The generic files are typically reviewed and edited using ArcGIS. If this is the case, the dBase part of the ArcView shape file should be converted back to a tab-delimited format using Excel or some other format conversion tool prior to executing the TransimsNet program.

TransimsNet generates complete TRANSIMS node, link, activity location, parking, process link, lane connectivity and pocket lane files. It also generates sign and signal warrant files that could be used as input to the synthetic traffic control process. As shown in Figure 4, the user can control the TransimsNet processing in a number of ways. There are control parameters the user can manipulate to adjust the generation of pocket lanes by functional class, signal warrants by area type and functional class combinations, and the access frequency (i.e., parking lots and activity locations) by area type.

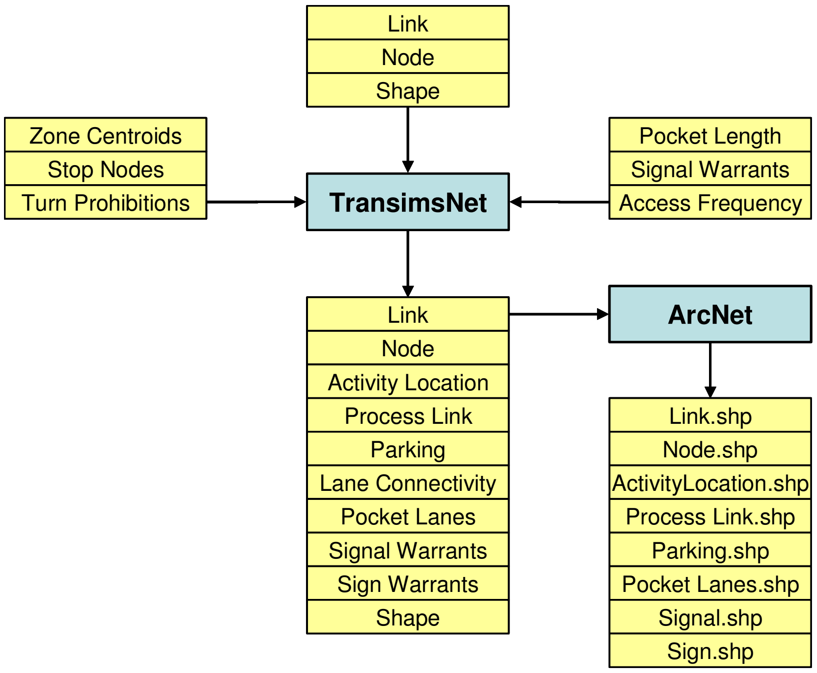


FIGURE 4: Synthetic Highway Network

Three additional input files can be provided. The zone centroid file includes the coordinates for the zone centroids and the area type of the zone. The zone centroid is used to assign each activity location to the nearest traffic analysis zone. This is needed in the trip table conversion process. The area type is used to control the signal warrants.

The user may also provide a list of turn prohibitions. These records override the default lane connectivity logic at a given intersection. If there is no lane connectivity between two links, the traffic movement is prohibited within TRANSIMS. For this application, the 400 turn prohibitions coded for the trip-based model were used as input to TransimsNet.

The stop nodes file is used to override the mid-block node collapsing logic. Normally, the software will eliminate mid-block nodes that were included in the regional model as zone access locations. Since the zone connectors are not used by TRANSIMS, the extra nodes are no longer needed. The TransimsNet program eliminates these nodes in order to make the network processing more efficient. This can create problems for the transit network conversion if the node is also used as a rail station or the end point of a bus route. In these cases, it is important to keep the node in the highway network. The stop node file enables the user to specify nodes that should not be eliminated.

Figure 5 shows the synthetic TRANSIMS network created from the Portland METRO regional model. Figure 6 shows the pocket lanes that were added at arterial intersections and freeway ramps. Figure 7 shows an example of the activity locations, parking lots, and process links added to the network for trip access. These replace the zone centroids and zone connectors used in the EMME/2 model. Figure 8 shows the lane connectivity details generated for a typical four legged intersection.

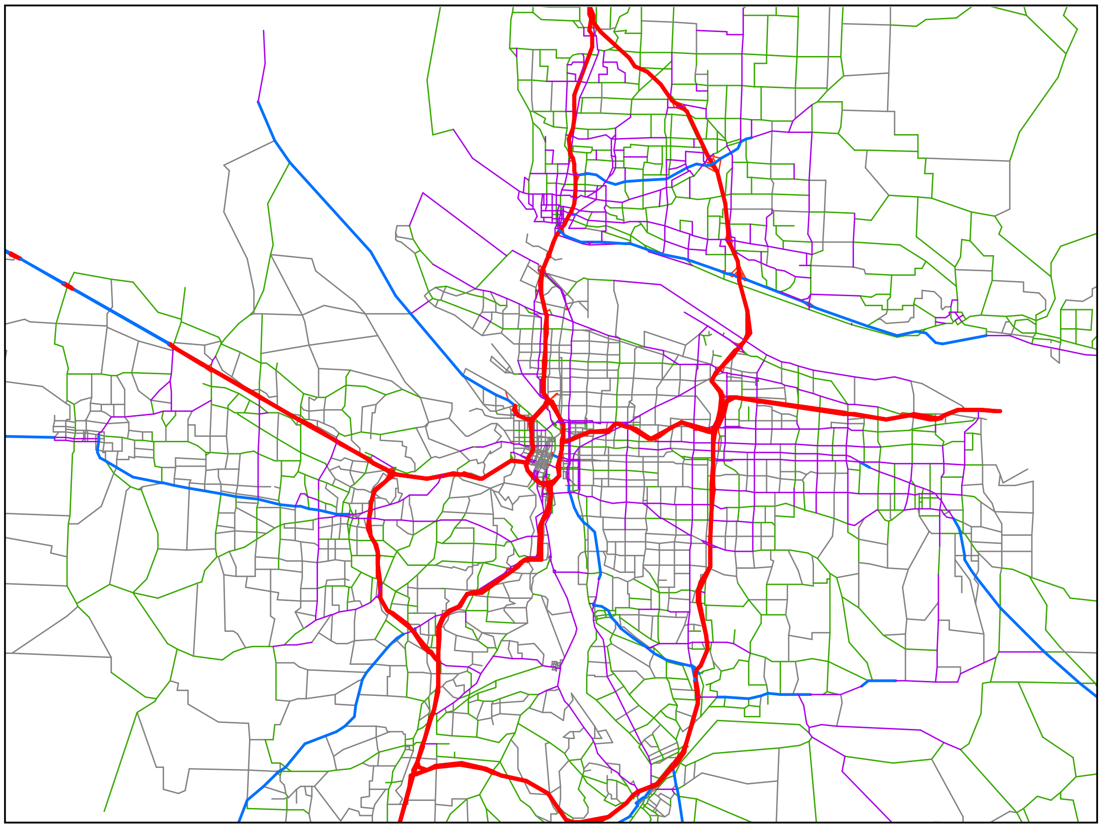


FIGURE 5: Portland TRANSIMS Highway Network

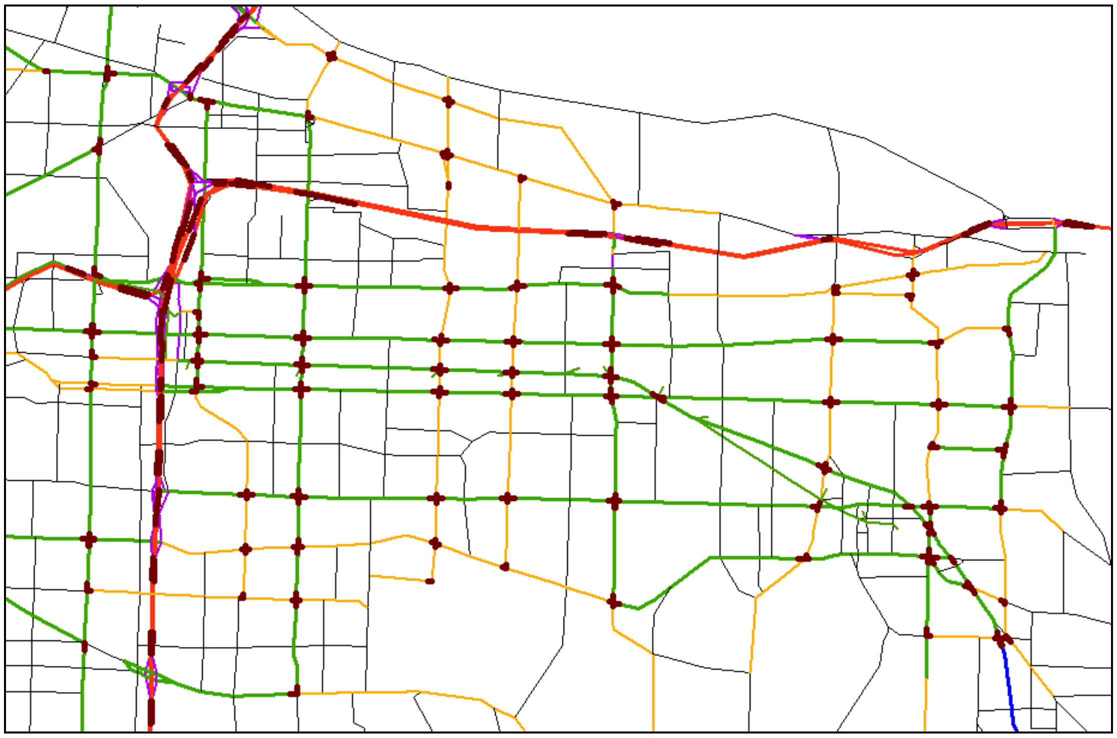


FIGURE 6: Pocket Lane Locations

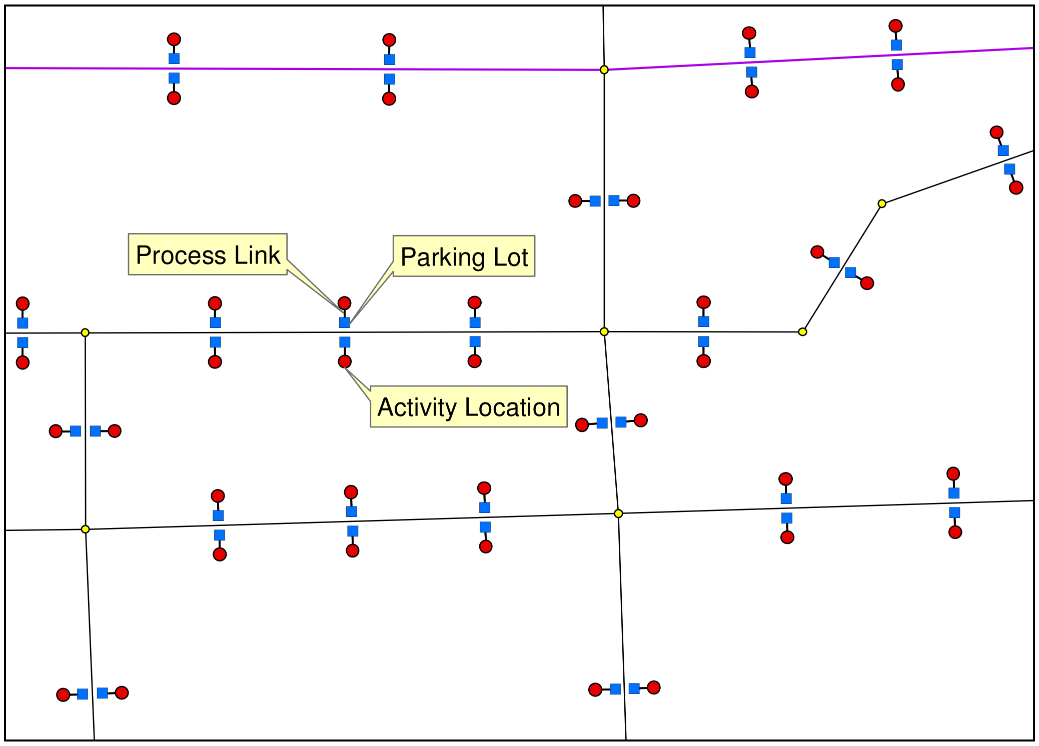


FIGURE 7: Network Access Coding

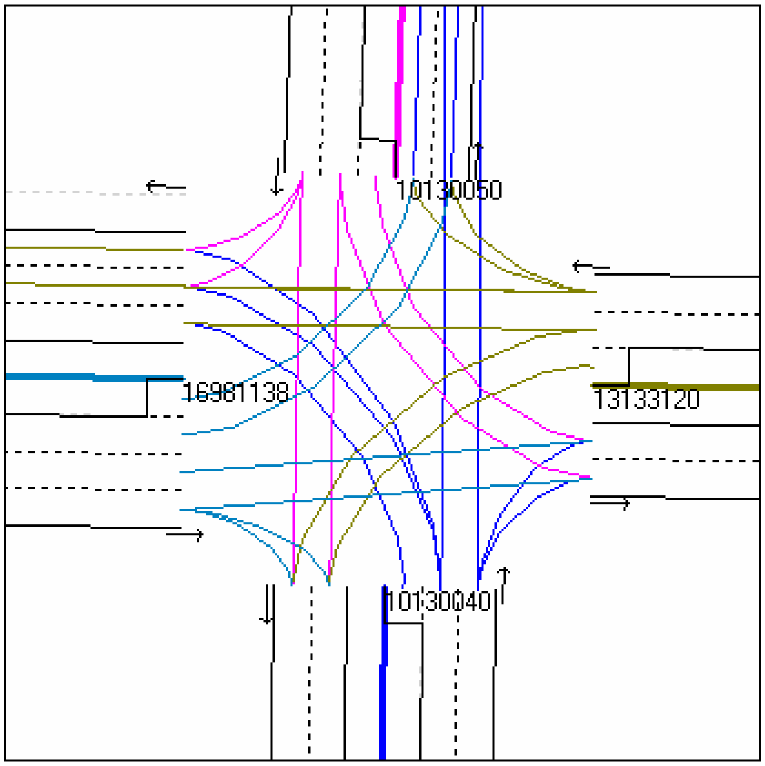


FIGURE 8: Lane Connectivity Coding

2.2.2 Traffic Control Warrants

In addition to basic network information, the TransimsNet program creates sign and signal warrants. The warrants are based on the facility types entering the intersection, the number of approach and departure links, and the area type assigned to the intersection. The warrant files can be converted to ArcView shape files using the ArcNet program and reviewed within ArcGIS. Typical intersection warrants are shown in Figure 9. These include several signalized and four-way stop intersections as well as intersections with stop signs on the minor approach, but no control on the major street.

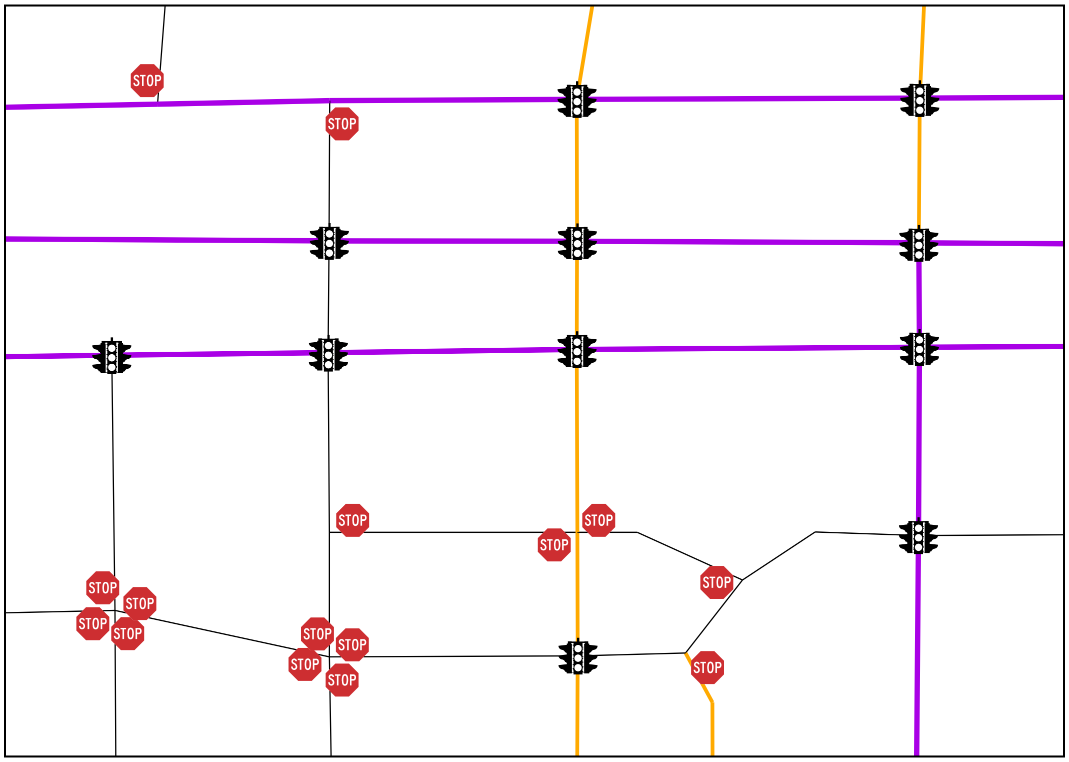


FIGURE 9: Intersection Control Warrants

As part of the network development effort for the "All-Streets" network, Portland METRO developed a signal inventory file based on information provided by the City of Portland and other sources. This information did not include all of the signals for all of the jurisdictions within the region, but did provide a useful point of comparison for calibrating the traffic control warrants. The signal inventory shown in Figure 10 was compared to the signal warrants generated by the TransimsNet program. Adjustments to the facility type by area type relationships were made to improve the signal warrant criteria and the program was re-run. Approximately ten iterations were needed to generate an 85 percent match. This included mismatched controls at intersections that did not exist in the MPO-level network. Most of these represent special purpose signals at parking lot entrances to Malls, major employers, and schools.

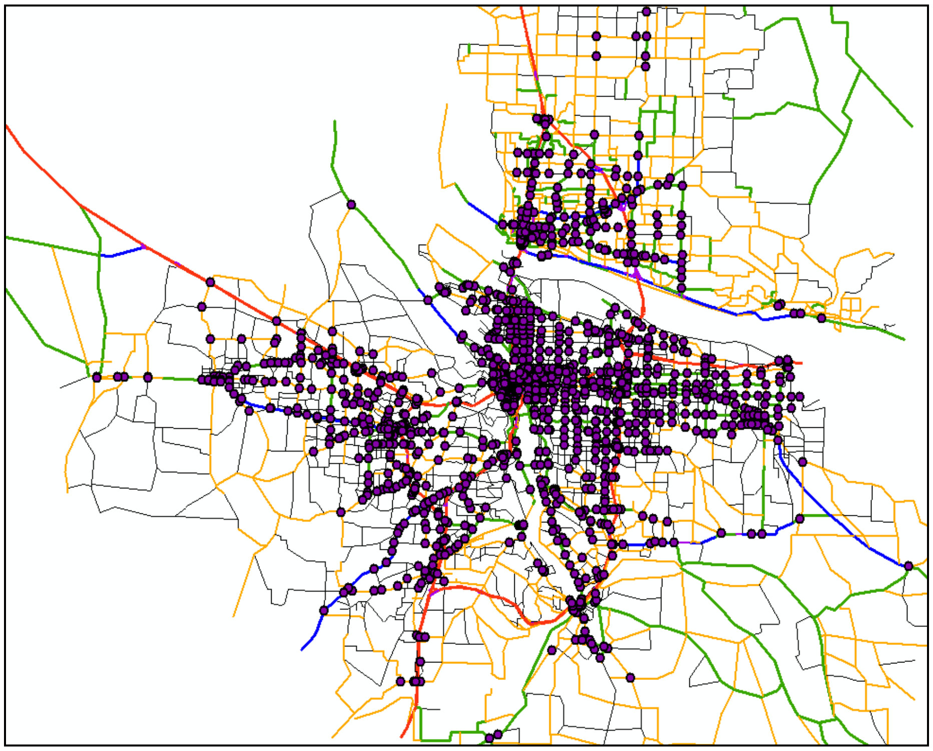


FIGURE 10: METRO Signal Inventory

The resulting signal warrant area types are shown in Figure 11. Seven sets of area type warrants were used. The signal warrants associated with these area types were as follows:

1. Collectors and Local Street
2. Collectors and Collectors
3. Minor Arterials and Collectors
4. Minor Arterials and Minor Arterials
5. Major Arterials and Minor Arterials
6. Major Arterials and Major Arterials
7. None

The area type definitions used for the Portland study are as follows

1. Central Business (pink)
2. Urban Business (red)
3. Urban Residential (dark green)
4. Suburban Business (medium green)
5. Suburban Residential (light green)
6. Rural Residential (yellow)
7. Rural (orange)

For example, if the intersection is located in area type 3 (Urban Residential Area), and the primary intersecting roadway is a minor arterial or above and the secondary roadway is a collector or above, the intersection warrants a traffic signal. Part of the reason the area type map appears a bit distorted relative to the development density within Portland and the state of Washington is an outgrowth of the coding differences included in the EMME/2 network. (These differences are described in section 2.2.1).

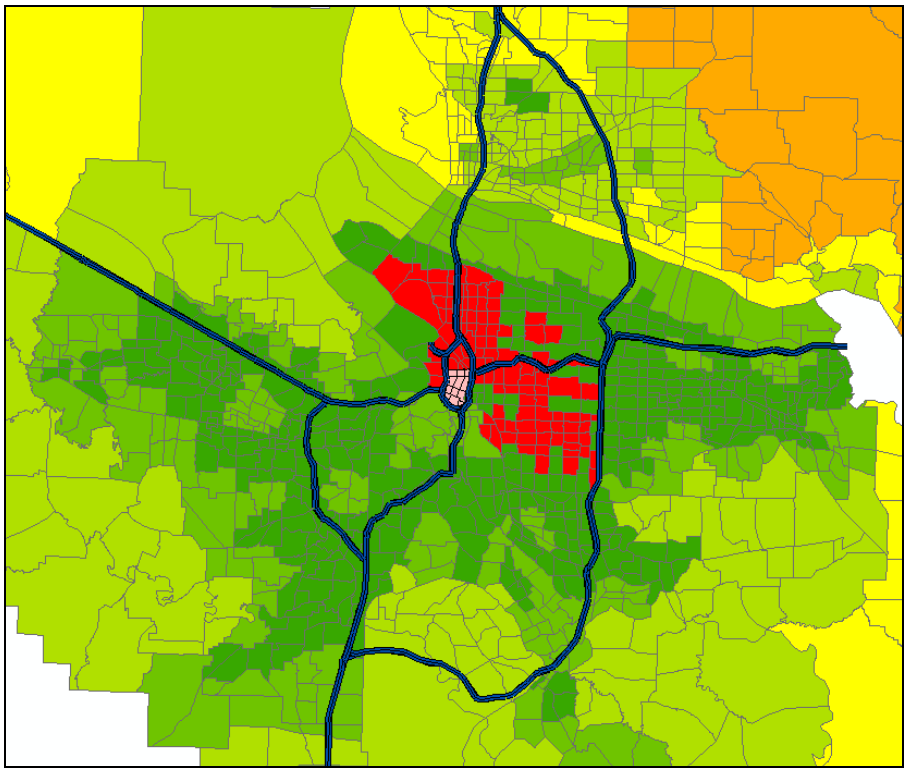


FIGURE 11: Traffic Signal Area Types

A comparison of the resulting signal warrants to the signal inventory file is shown in Figure 12. The signal and sign warrant files were converted to ArcView shape files using the ArcNet program and reviewed with ArcGIS. METRO made a number of adjustments and corrections based on local knowledge and overlaid aerial photos.

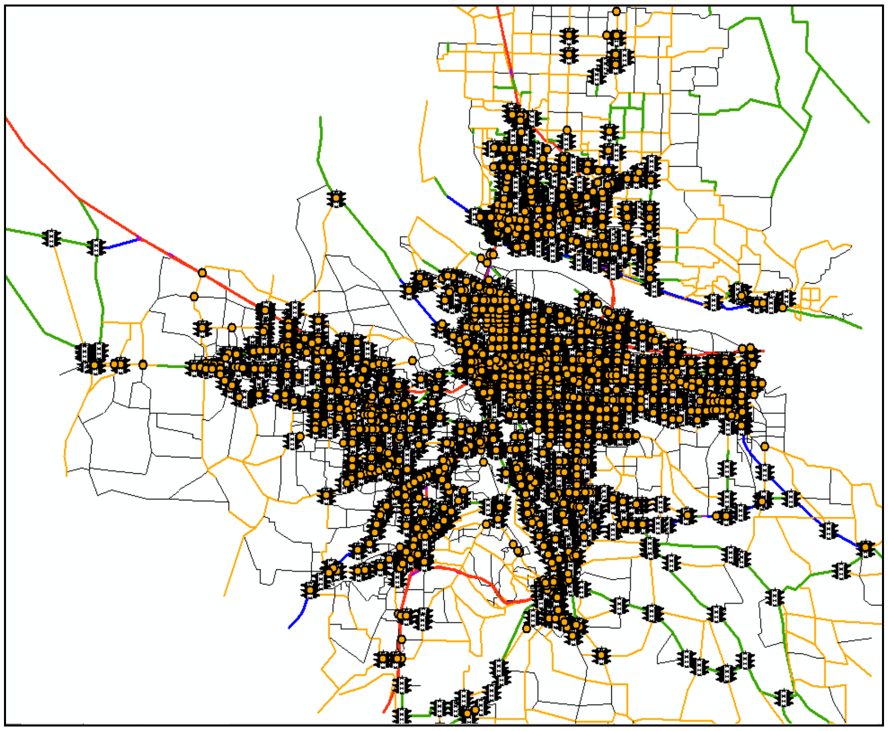


FIGURE 12: Synthetic Signals vs. Signal Inventory

2.2.3 Synthetic Traffic Controls

The traffic controls required by the TRANSIMS Microsimulator were synthesized using a two step process. In the first step, the IntControl program was used to generate the synthetic TRANSIMS signal control files using the reviewed and edited signal and sign warrant files generated by the TransimsNet program. This results in a set of unsignalized node, signalized node, timing plan, phasing plan, detector, and signal coordinator files in TRANSIMS format. The second step is optional. It creates signal offsets for coordinated fixed timed signal systems.

By default, the IntControl program creates demand actuated signals. The user has the option of specifying the areas within the region where fixed timed signals are created. For the Portland study, all the signals in zones in the Central Business District or Urban Business Districts were set to fixed timed signals. The Progression program was then used to calculate signal phase offsets for a sequence of fixed timed signals based on link travel times. The Progression program sets the signal offsets to minimize the number of times a vehicle must stop on the fixed timed network. Figure 13 shows how the IntControl and the Progression programs were used and some of the parameters that are available to control the process.

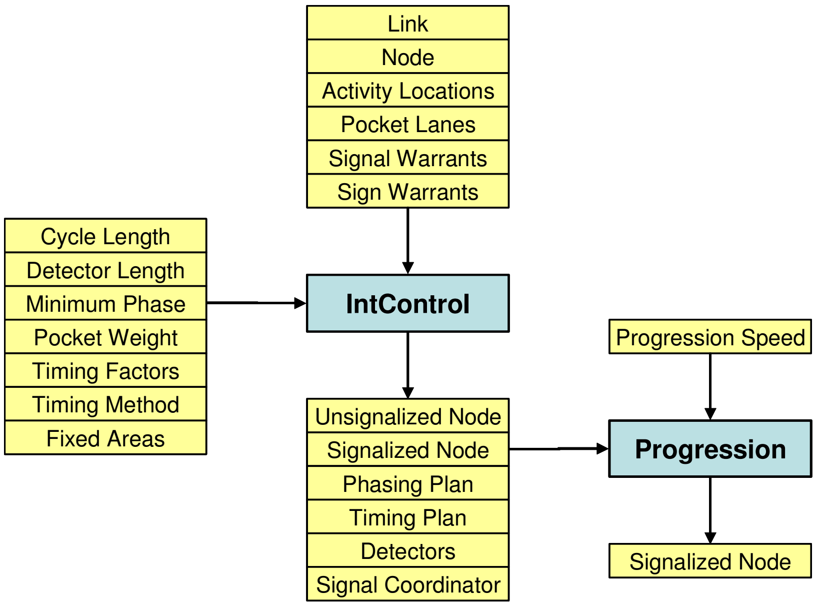


FIGURE 13: Synthetic Traffic Control Generation

For signalized nodes, the IntControl program uses the approach and departure links and the pocket lane information to determine the phasing plan for the intersection. For a given approach the software identifies the thru link, the opposing link, and left and right turn links. When left turn pocket lanes are coded on the approach and the opposing approach, the program assigns a protected left turn phase for the two left turns followed by an unprotected phase for the thru movements. When left turn pockets are present on only one of the approaches, the software assigns a protected left and thru phase followed by an unprotected phase for the opposing approach. If no left turn pockets are present, the software assigns an unprotected phase for both approaches. The software also assigns "right on red phases" if the cross-street has protected phases.

After determining the number of phases, the timing plan for each phase was estimated based on the lane capacities. The capacity assigned to each link is divided by the number of lanes to calculate the lane capacity of the approach. The lane capacity value is multiplied by the number of lanes included in each phase to estimate the weighting factor for the phase. A user-provided pocket lane factor is applied to the lane capacity when the lane is a pocket lane. This typically reduces the impact of a pocket lane on the overall phase capacity. For phases that include opposing approaches, the phase capacity is the average capacity of the two approaches. If the computed phase capacity is less than a user-provided minimum value, the phase capacity is increased to the minimum value. This effectively represents a minimum green time for any phase. The signal cycle length is then distributed proportionally to each phase based on the phase weighting factors and the yellow and all-red time parameters.

The TRANSIMS Network Editor was used to review the phasing and timing plans. If the review suggested that an overall adjustment to the timing plans was desirable, the necessary modifications were made to the IntControl parameters and the process was rerun. If intersection specific modifications were desired, the changes were made using the Network Editor.

2.2.4 Signal Progression

After the synthetic traffic signals were generated and reviewed, the Progression program was used to calculate the signal phase offset for each fixed timed signal in the dataset. The program sets the signal offsets to minimize the number of times a vehicle must stop on the fixed timed network. The Progression program begins by organizing the fixed timed signals into arterial groups based on the thru street designation on the link file. An arterial group is a sequence of fixed timed signals for a unidirectional movement bound between demand-actuated signals or stop controlled nodes. An example of an arterial group is shown in Figure 14 below:

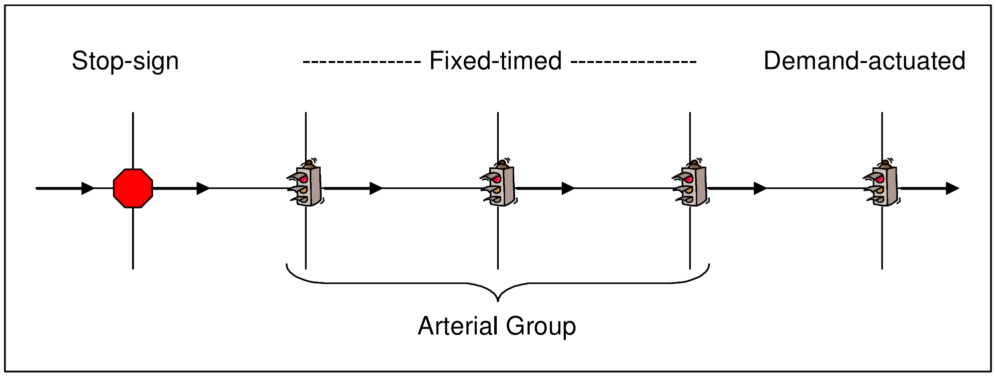


FIGURE 14: Signal Progression - Arterial Group

The algorithm sets the phase offset for the first signal in one of the arterial groups to zero and then uses the link travel times or a user-provided signal progression speed to calculate the signal offset for each subsequent signal in the group. The offsets are set so that a vehicle traveling at the designated speed could travel the full length of the arterial group without stopping. Once the offset of each signal on the first arterial group is set, the offsets on each cross street are calculated. This process continues until all of the groups are set. The program then sums up the percentage of each thru movement that could be traversed without stopping. This becomes the score assigned to the progression plan. The process is repeated using each of the other arterial groups as the first group. The combination with the highest score is selected as the best progression plan.

If desired, the user can manually edit the signalized node table to override the offset calculations at certain intersections. If the Progression program is re-run, it attempts to optimize the offsets for the rest of the signals based on the user-provided values. For the Portland study, all the signals in the CBD were made fixed timed signals with a signal progression speed equal to 10 mph. Free flow speeds were used to estimate the signal offsets in other areas. Figure 15 shows the fixed and actuated signal designations near downtown.

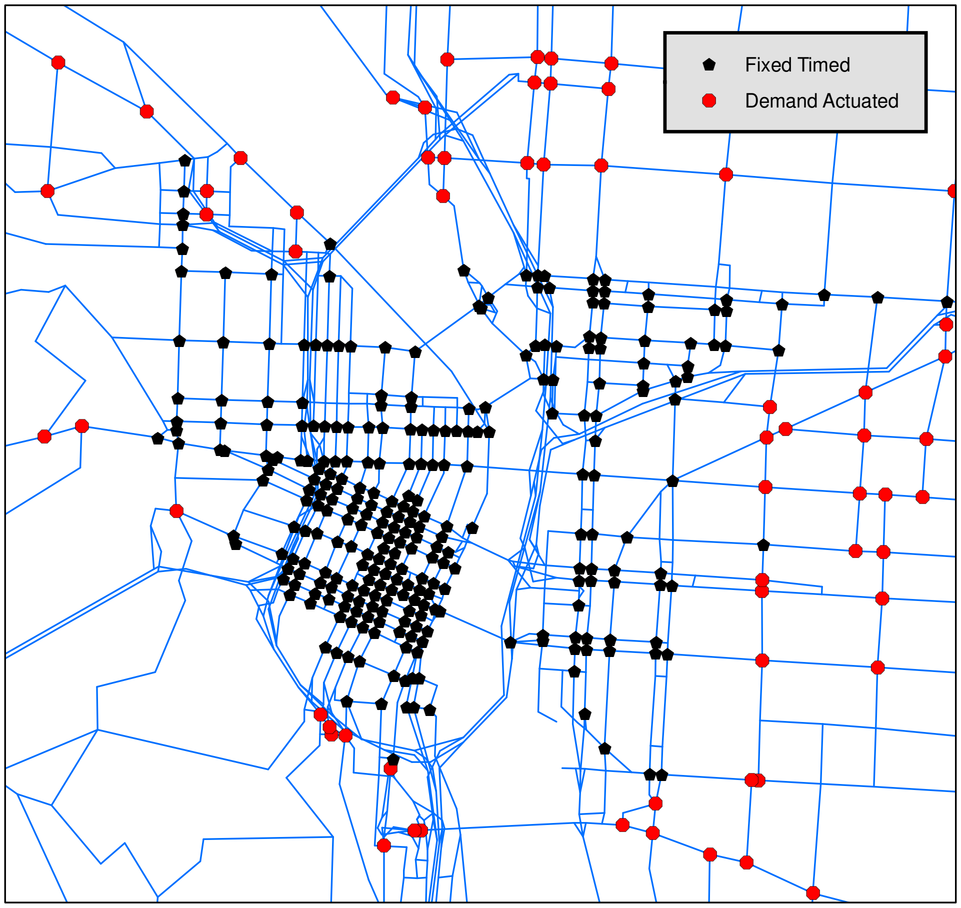


FIGURE 15: Synthetic Traffic Controls

2.2.5 TRANSIMS Transit Network

The TransitNet program was used to convert the generic route header and the route node files generated by the Emme2Route program to the transit network files required by TRANSIMS. These files include the transit stop, transit route, transit schedule, transit driver plan and transit vehicle files. It also uses or updates the link, node, activity locations, process links and parking files generated by the TransimsNet program. Another important input file contains a list the park-and-ride nodes. The user may also specify the maximum distance between transit stops by area type. Figure 16 shows how the TransitNet program was used to generate the TRANSIMS transit network.

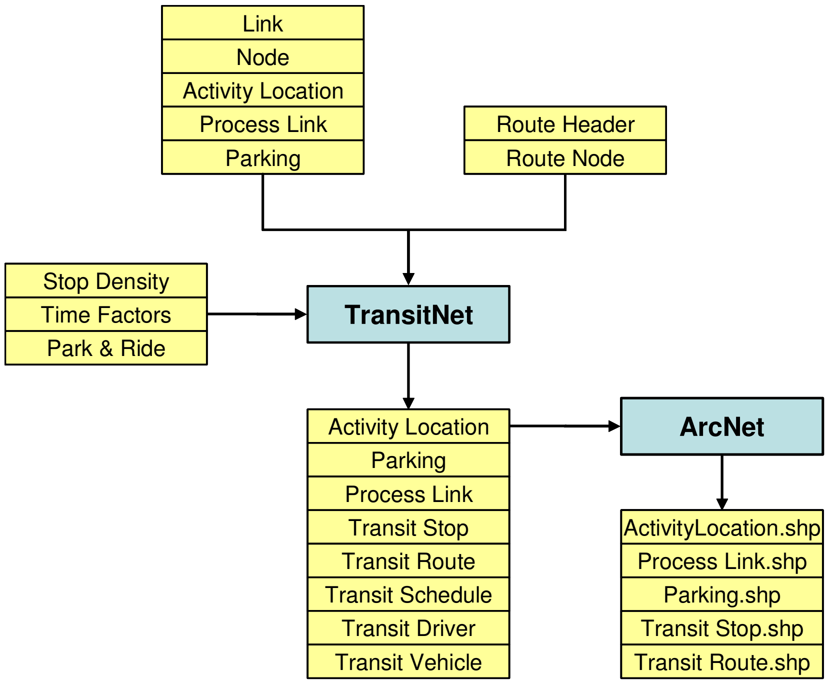


FIGURE 16: Synthetic Transit Network Generation

As described in section 2.2.4, the route header file includes headways and offsets for different time periods. The route node file lists the network nodes along the route plus the dwell times and travel times between stops. TRANSIMS, on the other hand, is designed to represent the travel path of each vehicle and the actual location and time of day when the vehicle stops to serve passengers. It also needs to know the origin and destination activity location of the driver and the origin and destination parking location of the vehicle.

The TransitNet program starts by synthesizing transit stops on all links in the network that permit bus or rail access. If the link length is greater than twice the user-defined maximum stop spacing for the corresponding area type, the link is split into segments and a near-side stop is added at the end of each segment in each direction. The routes are then assigned to the links and the appropriate stops are selected for each route. The route mode and the dwell time values determine which stops are selected for a particular route. Rail modes only stop at the locations closest to the network node designated as a stop. Bus routes stop at each synthetic stop location on the links where the network nodes at both ends are designated as a stop. If the route makes a left turn at a multi-lane intersection, the near-side stop before the intersection is excluded from the route. This is needed to give the bus enough distance to maneuver into the left lane before the required turn.

After the stop list is generated for a given route, the route schedule is synthesized. The transit schedule file lists the time of day when each run of a given route visits each transit stop on the route. The offset value determines when the first run of the route starts from the first stop. The travel time factored by a time of day adjustment factor plus the dwell time for each stop is used to establish the departure time for each successive stop. The start time for the next run is based on the headway and the time period duration.

Each run also requires a driver plan and a transit vehicle. The TransitNet program generates a separate driver and vehicle for each run of each route. The origin and destination of the driver and vehicle are selected from the activity locations and parking lots on links that approach the first transit stop and links that depart from the last transit stop. If possible, the program will attempt to select a parking lot on a thru link. If a thru link is not available, a right turn maneuver is preferred over a left turn maneuver. The selected link must also permit access to bus or rail vehicles.

The TransimsNet program generates parking lots on links that are auto accessible. If a network link is coded as transit or walk only, parking lots would not be added to the link. Since TRANSIMS requires that each transit route start and end at a parking lot, routes that end on dead end links or exclusive guideways require some coding adjustments to properly synthesize. In situations where a U-turn is required to access a parking lot, U-turn lane connectivity was added to the end link. In some cases, the vehicle restriction on a link attached to the end node was modified to permit bus or rail vehicles. In other cases, an extra link was added at the end of the line in order to add a parking lot and activity location for the transit route.

Adjustments were also needed on loop routes. TRANSIMS does not permit a route to stop at the same transit stop twice. If the loop was at the end of a route, the simplest solution was to change the location of the last stop. The route could end one stop early or proceed to one stop beyond the original end of the route. If the loop was in the middle of a route, it was generally best to alternate stop locations over the duplicate section. In some critical locations it was best to add extra bus stops to keep the routes separate, but continue to provide equal access to both legs of the loop.

Some park-and ride lots required extra attention as well. The TransitNet program uses the network nodes listed in the park-and-ride file to identify the closest parking lot on one of the links attached to each node and changes to parking lot flag to permit park-and-ride access. In order to be effective, however, a traveler must be able to drive to the parking lot and walk from the parking lot to the transit stops that provide park-and-ride service. In some cases, the drive and walk access restrictions needed to be adjusted to facilitate the connection. Walk access restrictions also affected rail stations on exclusive guideways. Since walking was not permitted on the rail guideway, special access links were needed to attach the rail station stops to the near-by street network.

The TransitNet program is designed to display useful error messages when it encounters issues with the way the transit routes are coded. The user typically runs the process several times to identify and correct the route coding or enhance the underlying network links. Visualizing the issues in ArcGIS often helped to identify appropriate solutions. Figures 17 and 18 show examples of the Portland transit network in TRANSIMS format as depicted by ArcNet. It took METRO about two weeks to debug the transit coding.

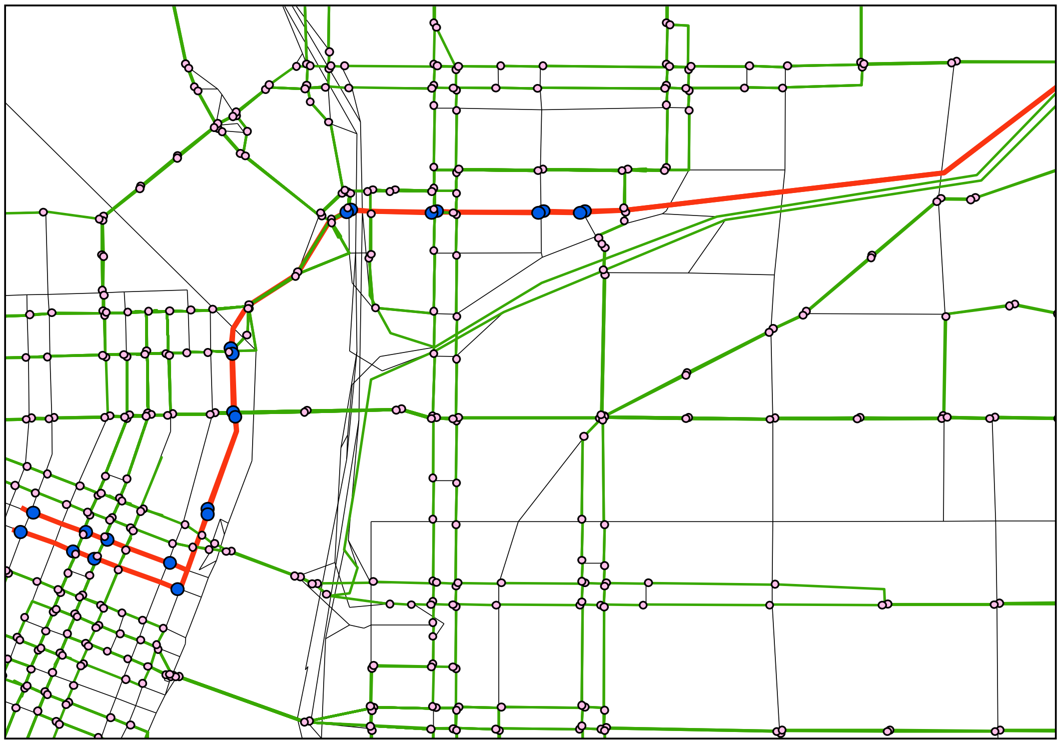


FIGURE 17: Synthetic Transit Network

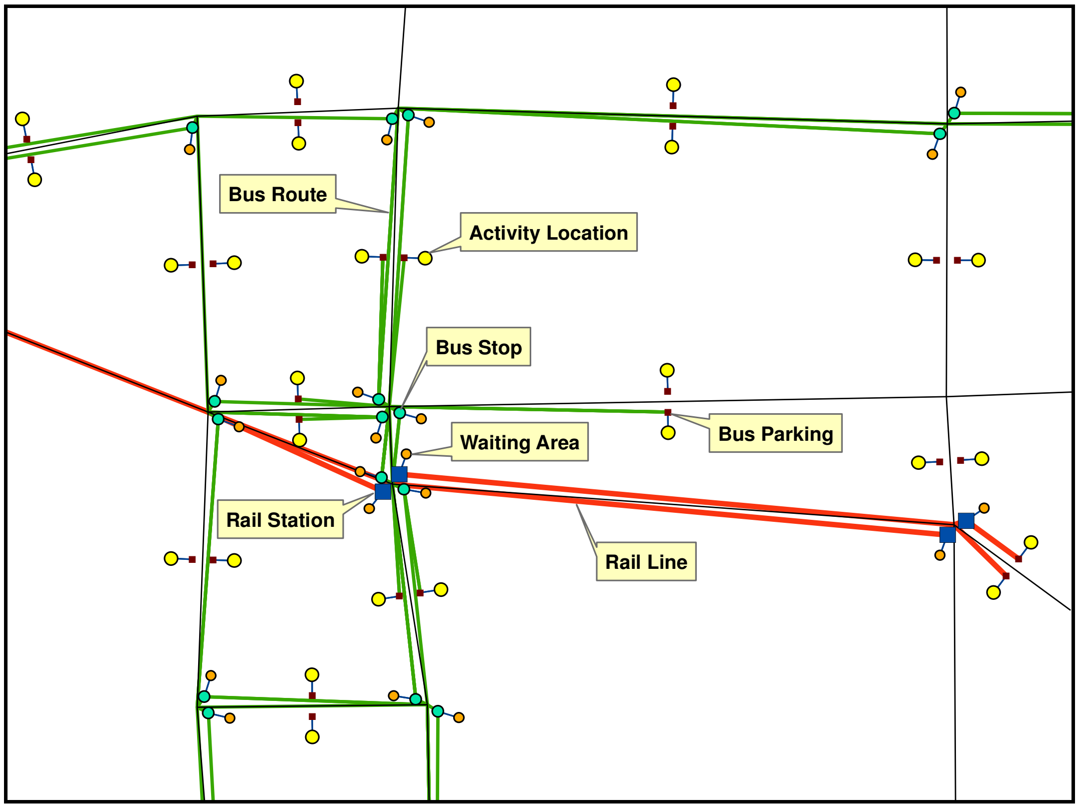
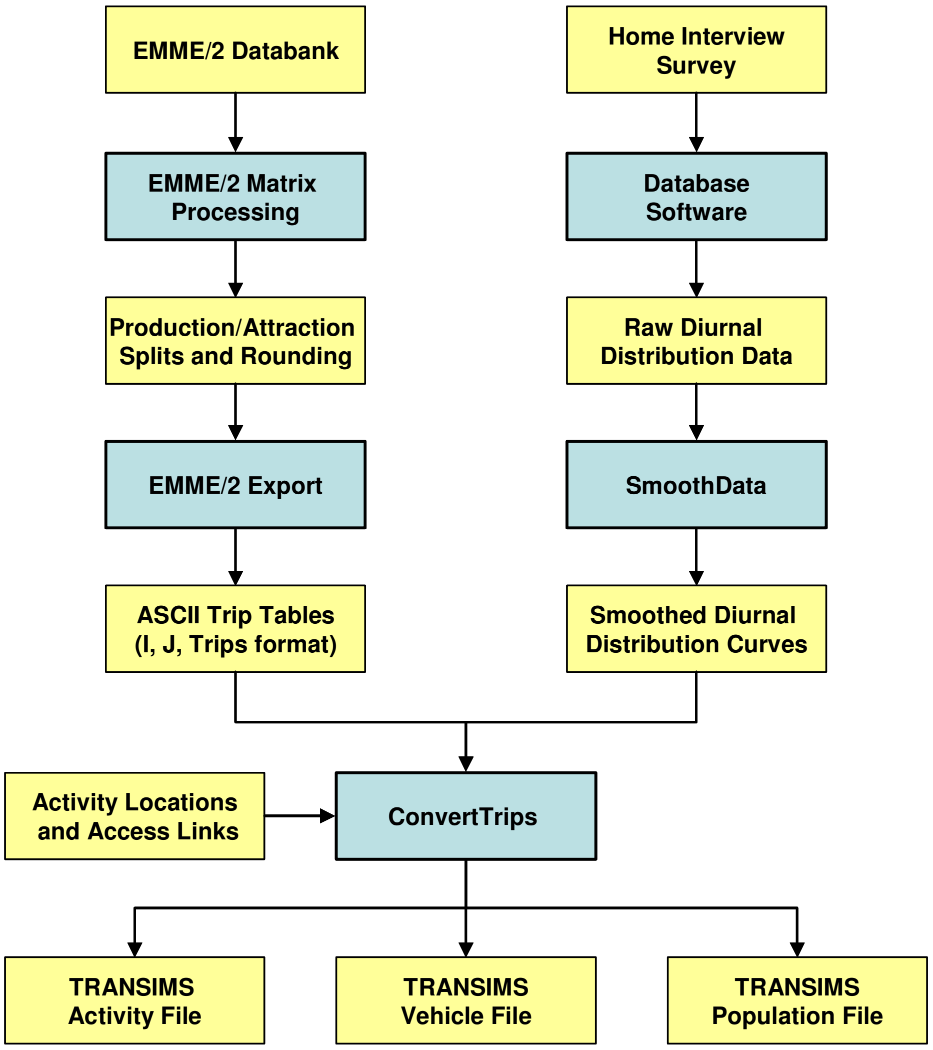


FIGURE 18: Transit Coding Details

Chapter 3 Trip Table Conversion

The process used to convert daily trip tables from METRO's regional model to the activity data expected by TRANSIMS is summarized in Figure 19. As the diagram suggests, METRO reformatted the data within EMME/2 before exporting the information in ASCII format. They also estimated diurnal distribution curves using their household travel survey and then smoothed the results to account for response bias. These results were then input into the ConvertTrips program to generate the TRANSIMS Activity, Vehicle, and Population files. For the Portland study, each trip in the METRO trip table was converted to a single person household making one trip during the day. This results in two activities in the Activity file. The first activity is to stay at "home" at the origin activity location until the trip begins. The second activity is to start working or shopping at the trip destination activity location based on the estimated travel time between the origin and destination activity locations. The person remains at the destination activity location for rest of the day. The METRO base year vehicle trip tables include about five million trips. In TRANSIMS format, this translates into five million households and ten million activities.



EMME/2 Databank
EMME/2 Matrix Processing
Production/Attraction Splits and Rounding
EMME/2 Export
ASCII Trip Tables (I, J, Trips format)
Home Interview Survey
Database Software
Raw Diurnal Distribution Data
SmoothData
Smoothed Diurnal Distribution Curves
Activity Locations and Access Links
ConvertTrips
TRANSIMS Activity File
TRANSIMS Vehicle File
TRANSIMS Population File

FIGURE 19: Trip Table Conversion Process

3.1 Data Preparation

3.1.1 Trip Tables

Data from Portland METRO's 1994 regional model was used for this analysis. The average weekday vehicle trip tables consist of six internal trip purposes:

1. Home Based Work
2. Home Based Other
3. Non-Home Work
4. Non-Home Non-Work
5. School
6. College

and five external trip purposes:

1. Home Based Work (External to Internal)
2. Home Based Non-Work (External to Internal)
3. Recreation (Internal to External)
4. Non-Recreation (Internal to External)
5. External to External

and one truck trip table. The truck trip table is in Origin-Destination format and the eleven other tables are defined in Production-Attraction format. All of the trip tables represent zone-to-zone trips as floating point numbers. There are 1260 zones in the METRO model.

3.1.2 Production-Attraction Splits

Each of the eleven daily trip tables defined in Production-Attraction format were split into production-to-attraction trips (PA) and attraction-to-production trips (AP) using factors obtained from the household travel survey. The attraction-to-production trips were then transposed in order to make the attraction zone the trip origin and the production zone the trip destination. Each direction was then stored as a separate trip table. In other words, the daily Home-Based Work production-attraction trip table was split into Home-Based Work trips from the production zone to the attraction zone (i.e., home to work) and Home-Based Work trips from the attraction zone to the production zone (i.e., work to home). This resulted in a total of 23 trip tables used in the conversion process.

Since METRO trip tables use floating point numbers and TRANSIMS needs discrete trips, the trip tables were integerized before they were exported. METRO used a bucket rounding procedure to generate the integer trip tables. These trips were then exported to ASCII format for input to the conversion utility. The input trip table file is a space or tab-delimited ASCII data file with the three data fields shown below:

FROM_ZONE	The origin zone number of the trips.
TO_ZONE	The destination zone number of the trips.
NUMBER OF TRIPS	The number of trips (integer format).

If the input trip table files include header records, they are ignored. A sample file is shown below:

```
1       1       3
1       2       100
```

1	8	1
1	10	11
1	12	2
2	1	5

3.1.3 Diurnal Distribution

The daily trips are randomly assigned to a specific start time using diurnal probability distributions. A separate diurnal distribution was used for each of the 23 trip tables. Each of the internal diurnal distributions was derived from METRO's household travel survey. METRO developed 15-minute diurnal distributions for the twelve internal trip types. For the external and truck trip tables the percent distributions were estimated for five time periods (Midnight to 7:00 AM, 7:00 AM to 9:00 AM, 9:00 AM to 4:00 PM, 4:00 PM to 6:00 PM and 6:00 PM to Midnight). The trip diurnal distribution file is a space or tab-delimited ASCII data file with the three data fields shown below:

START-TIME	The hour of the day at the beginning of the probability increment
END-TIME	The hour of the day at the end of the probability increment
PERCENT	The percentage of trips or the probability value for the time period.

If the diurnal distribution files include header records, they are ignored. A sample diurnal distribution file developed for Home-Based Work internal-to-external trips is show below:

0	7	0.1321
7	9	0.3975
9	16	0.3105
16	18	0.0278
18	24	0.1321

A portion of the diurnal distribution file developed for Home-Based Work production-to-attraction trips with 15-minute increments is shown below:

0	0.25	0.00030
0.25	0.5	0.00027
0.5	0.75	0.00026
0.75	1	0.00025

3.1.4 Smoothing Diurnal Distributions

The diurnal distribution curves derived from METRO's home interview survey included considerable response bias (e.g., reporting start times to the nearest 15 minutes). This resulted in diurnal distributions by 15-minute increments that were high on the hour and half hour and low at the quarter hours. Figures 20 and 21 show an example of the raw Home-Based Work and Home-Based Other distributions. When these distributions were used in the trip conversion process, the resulting simulation had difficulty loading the large surge in traffic at the beginning of each hour.

To address this issue and compensate for the response bias in the survey, the diurnal distributions were smoothed using the SmoothData program. This program uses multiple iterations of a moving average technique to smooth the raw diurnal distribution data. A three record moving average (current record, previous record and the next record) was applied ten times to produce the smoothed distributions used in the trip conversion process. The results of this smoothing process are also shown in Figures 20 and 21.

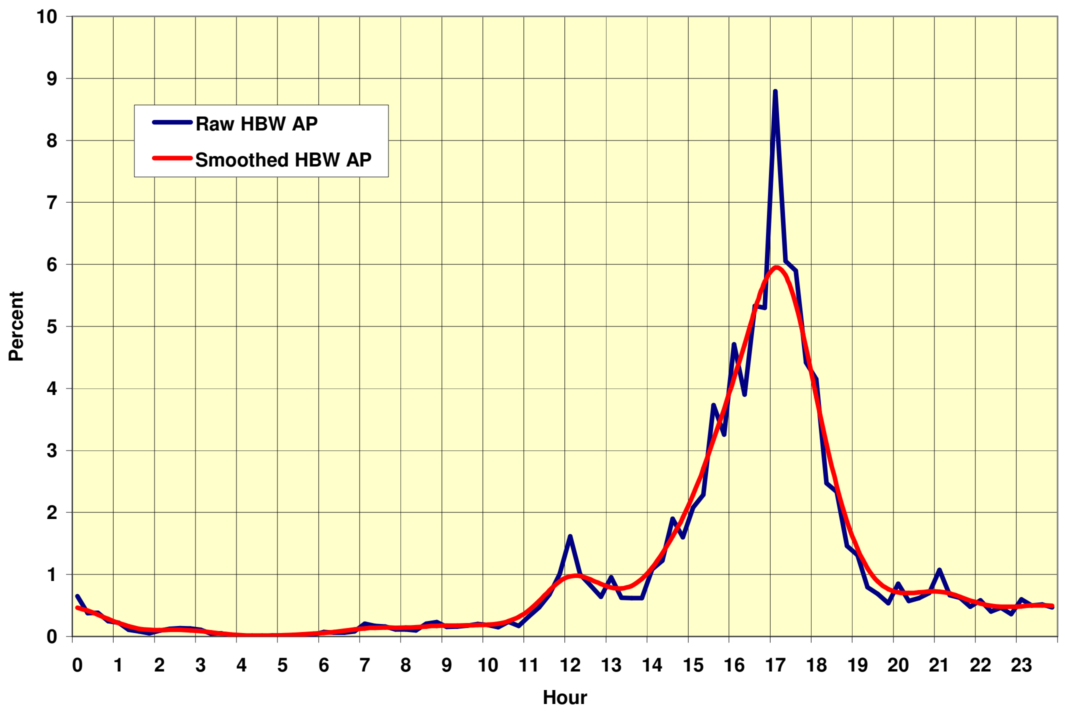


FIGURE 20: Home-Based Work Diurnal Distribution

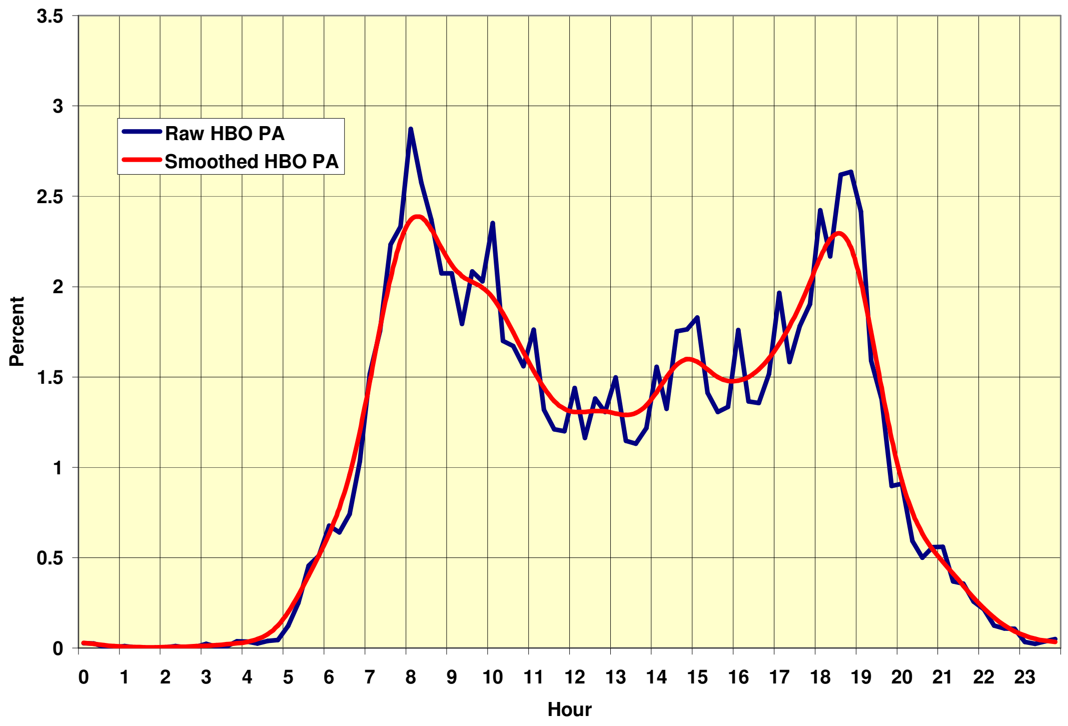


FIGURE 21: Home-Based Other Diurnal Distribution

3.2 Conversion Process

Once the zone-to-zone trip tables and smoothed diurnal distribution curves were prepared, the ConvertTrips program was used to generate the TRANSIMS activity, vehicle, and population files. The activity file defines the time of day, duration, and location of each activity a person is engaged in over the course of the day. It also defines the mode of travel and vehicle the person will use to travel from one activity location to another. The vehicle file defines the initial location of each vehicle and the vehicle type. The population file includes information about each household and each person within the household.

For this application, the ConvertTrips program assigned each trip in the trip table to a specific activity location within the origin and destination zones and selected the time of day when this trip begins. A household and person record was generated for each trip and a vehicle was generated for automobile and truck trips. This section describes this process.

3.2.1 Network Data

Activity locations are defined within the TRANSIMS network as X, Y coordinate points. They typically represent the block face of each arterial roadway within the network. In other words, each link typically has one or more activity locations on each side of the street. Limited access facilities such as freeways and ramps do not generally include activity locations. Each activity location is attached to parking lots, transit stops, or other activity locations using access links defined in the Process Link table.

For the Portland study, activity locations were added to each side of the street at quarter mile spacing. Activity locations were not included on freeways, ramps, transit-only links, or bridges crossing the Willamette River. This resulted in about 20,800 activity locations for trip allocation. Each activity location was attached to a parking lot using two process links (one in each direction). Figure 22 shows a typical network configuration.

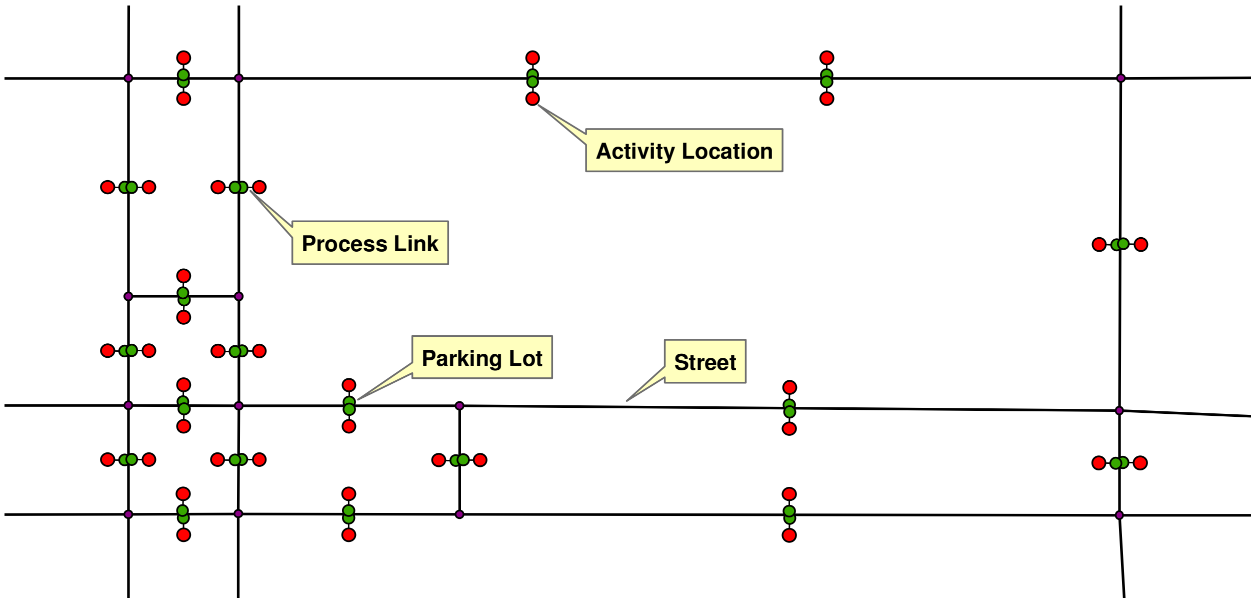


FIGURE 22: TRANSIMS Network Access

The activity location file permits the user to assign a number of user-defined fields to each activity location. In order to apply the ConvertTrips program, at least one user-defined field needs to be included in the activity location file. This field associates the activity location with the corresponding zone number. It is used to convert the zone numbers found on the input trip tables to activity locations in the TRANSIMS network.

In addition, most ConvertTrips applications use at least two additional user-defined fields. These fields define the relative attractiveness or weight assigned to a given activity location within the zone. If these fields are not defined, the software assigns equal probability to all activity locations within the zone. If fields are provided, the activity locations with higher values have an increased probability of being selected. Different user-defined fields can be provided for the origin and destination of a trip and for different trip purposes. This enables the user to more precisely locate trip ends to specific areas within a zone. For example, trips between home and work can select the origin location based on the number of households in the vicinity of each activity location within the origin zone and select the destination location based on the number of employees in the vicinity of each activity location within the destination zone. It is important to note that these weights only affect the allocation within a given zone. If all of the activity locations within a zone have the same value, the weighting factor has no affect on the results.

In Portland, equal weights were assigned to all activity locations within a zone with one exception. The Portland International Airport is in a relatively large zone that includes a minor roadway along the Columbia River (see Figure 23). In order to force the allocation process to select the activity locations that represent the airport terminal more frequently than the locations along the minor roadway, the weight assigned to the terminal activity locations was 10,000 times larger than the other locations within the zone.

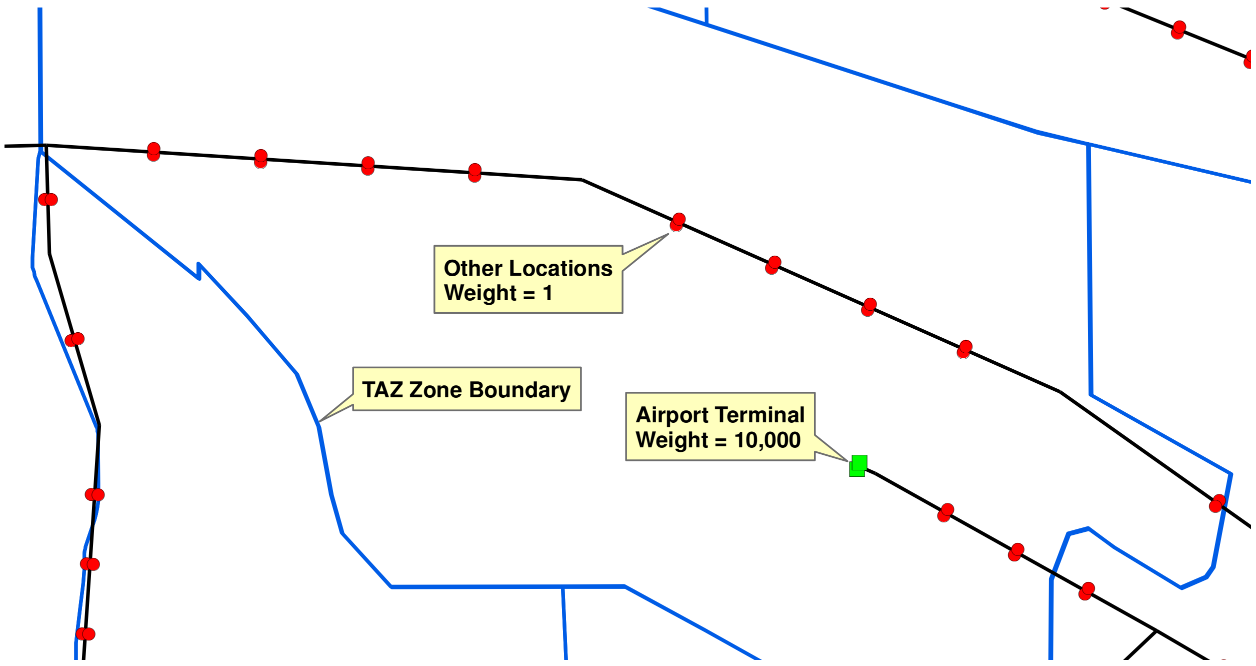


FIGURE 23: Airport Zone Trip Allocation Weights

3.2.2 Assigning Trips to Activity Locations

The conversion process reads each record in each trip table one at a time. The allocation process is then applied independently for each trip within the origin-destination pair. The process first assigns the origin of the trip to an activity location within the origin zone and then based on the origin location selects the destination location.

The origin location is randomly selected from the activity locations with non-zero origin weights assigned to the origin zone for this particular trip table. A cumulative distribution of the weights is generated for the zone and a random probability is used to select a location from the normalized distribution. Once the origin is selected, the cumulative distribution for the destination zone is generated. The weight assigned to each activity location within the destination zone is the product of the non-zero destination weights and the straight-line distance between the specific origin location and each location assigned to the destination zone.

Figure 24 provides a simplified example of the destination weighting process. In this example, zone 10 is the origin zone and activity location number 3 within zone 10 was selected as the origin activity location. Zone 20 is the destination zone and it contains five activity locations. The probability of selecting a destination activity location is the product of the distance from activity location 3 in zone 10 to each activity location in zone 20 and the destination weight for the activity location. Table 1 shows some typical calculations.

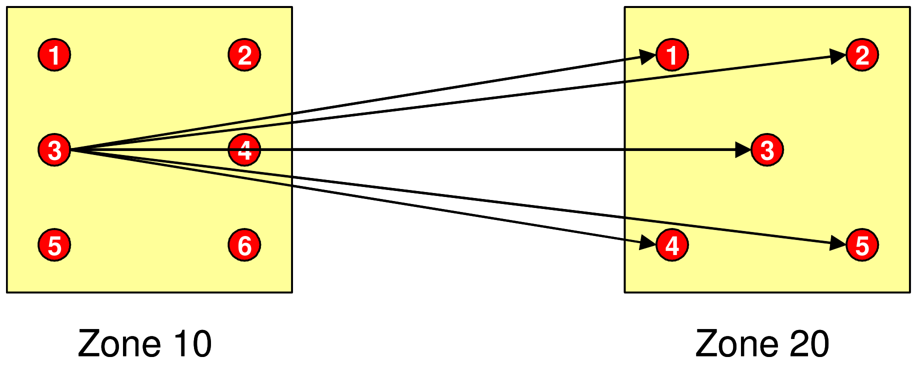


FIGURE 24: Destination Weighting Example

TABLE 1: Impact of Distance on Destination Weights

Zone 20 Location	Zone 20 Weight	Weight Distribution	Example 1 Distance to Origin	Example 1 Destination Weight	Example 1 Percent Probability	Example 2 Distance to Origin	Example 2 Destination Weight	Example 2 Percent Probability
1	10	16.7%	500	5000	6.8%	50000	500000	16.4%
2	15	25.0%	1950	29250	40.1%	51450	771750	25.4%
3	10	16.7%	1175	11750	16.1%	50675	506750	16.7%
4	15	25.0%	500	7500	10.3%	50000	750000	24.6%
5	10	16.7%	1950	19500	26.7%	51450	514500	16.9%
Total	60	100.0%		73000	100.0%		3043000	100.0%

Notice the impact of distance on the relative probability of selecting a given destination based on the Table 1 examples. Example 1 represents trips between near-by zones. In this case the probabilities are significantly affected by the relative distance between the origin and destination activity locations. Example 2 shows how the distance impact is almost non-existent for zones that are far apart. In other words, the distance weighting function favors destinations that are farther away when the relative trip length is short. Since this algorithm is primarily applied to vehicle trips, the implication is that very short vehicle trips are less likely. Another way to think about this is that people are more likely to walk if the distance is very short.

The impact of the distance weighting function is most significant when allocating intrazonal trips. The procedure tends to pick a destination as far away from the origin as possible. Notice also that the probability of selecting a single location as the origin and destination of the trip is zero. This is why it is important to include at least two activity locations within each zone. If the zone has only one activity location, all of the intrazonal trips within that zone will be lost. Of course, if the zone has no activity locations, no trips to or from that zone can be converted.

3.2.3 Determining the Activity Start and End Times

After determining the origin and the destination activity locations for a trip, the end time for the origin activity (i.e., the trip start time) is randomly assigned based on the diurnal probabilities of the associated trip table. The start time for the destination activity (i.e., the trip end time) is estimated based on the straight-line distance between the origin and destination activity locations and the user-provided average speed. The straight-line distance between the activity locations is divided by the average speed to estimate the travel time. For the Portland study the average speed was set at 16 meters per second (about 35 mph). To this value, a fixed time of nine minutes was added to account for trip end delays and slower speeds for shorter trips.

These two time values become the anchors for the two activity records added to the activity file. The first activity starts at time zero (midnight) and ends at the trip start time. The second activity begins at a time equal to the trip start time plus the estimated travel time and the trip end time. The end of the second activity is the end of the travel day. In Portland this was set at 27 hours (i.e., 3 AM the following day).

These values are not, however, entered into the activity file directly. The activity file defines activity times using upper and lower bounds. The time range was defined as a function of the trip purpose. Work trip arrival and departure times were defined with a 30 minute time window. Other trip purposes were given a 90 minute time window. In all cases, the two anchor times were assumed to be the center of the time window. For example, a work trip must arrive at work within plus or minus 15 minutes of the selected time. A shopping trip must take place within plus or minus 45 minutes of the selected time.

What this typically means is that TRANSIMS will start the trip at the desired time, but the arrival time is relatively flexible to account for wide variations in the simulated travel time. In other words, trips generated by this process are not likely to experience scheduling problems in the TRANSIMS Router or Microsimulator.

3.2.4 Other Attributes

In addition to the trip origin, destination, and start time, the ConvertTrips program allows the user to specify the trip purpose, the travel mode, and the vehicle type and subtype for each trip table. The trip purpose is saved in the activity type field of the activity file. The activity type is an integer number based on the following codes:

0. Home
1. Work
2. Shop
3. Visit
4. Social/Recreation
5. Other
6. Serve Passenger
7. School
8. College

The travel mode is the code corresponding to a mode string found in the Modes file. The default mode codes are:

1. Walk
2. Drive
3. Transit
4. Transit with Rail Bias
5. Park-&-Ride Outbound
6. Park-&-Ride Inbound
7. Bicycle
8. Magic Move (i.e., non-modeled travel – mostly used for auto passengers)
9. School Bus

The vehicle type and subtype codes are based on the Vehicle Prototype file. The default vehicle type codes are:

1. Automobiles
2. Trucks
5. Buses
8. Rail

A vehicle was added to the vehicle file for each trip contained within a trip table identified with a vehicle-based travel mode (i.e., Drive and Park-&-Ride). The vehicle was located at a parking lot attached to the origin activity location through a process link. For Drive trips, activity locations that are not attached to parking lots are excluded from the allocation process. In other words, Drive trips are only assigned to locations that have parking access at the origin and destination ends of the trip.

3.3 Refinement Tests

The process described above can be implemented relatively quickly and does not require detailed information about zones or sub-zone allocations. Since Portland METRO did have zone and sub-zone data, tests were conducted to determine the relative benefit of more detailed information on the analysis results.

3.3.1 Zone Number

The first step in the refinement process was to improve the assignment of activity locations to traffic analysis zones. By default, the TransimsNet program assigns each activity location to the closest zone centroid. In situations where the zone boundaries are irregular, this can frequently assign activity locations to the wrong zone number. Portland METRO had zone boundary polygons in ArcView shape file format. The point-in-polygon feature of ArcGIS was used to assign each activity location to the appropriate zone polygon.

This was an improvement over the shortest distance method, but it still had a number of problems. The first problem was that the EMME/2 highway network was not geocoded at the same level of detail as the zone polygons. This is partly due to the fact that EMME/2 uses straight-line links. It was also affected by the relatively low resolution coordinates used to digitize the EMME/2 nodes. Adding shapes to the network links was beyond the scope of this project, but coordinate refinement was feasible.

Since METRO generated a detailed GIS network for the All-Street application, the node coordinates from the All-Streets network were used to adjust the EMME/2 coordinates. The CoordMatch program was used to apply a rubber-sheeting methodology between the two TRANSIMS node files. A correspondence table was developed that associated a node number within the EMME/2 network with a node number in the All-Streets network. The coordinates for these nodes were replaced directly. For nodes that were not given a correspondence, the net coordinate shift based on the closest reference nodes in the North-South-East-West directions was applied.

The process of iteratively developing the node correspondence table and reviewing the network overlay on the zone boundary file took about one week to complete. Approximately 1980 nodes were included in the correspondence table. This represents almost one quarter of the nodes in the EMME/2 network. Even with this effort the match was not perfect. Figure 25 is an example of an area in Washington that still shows significant differences. In this case some of the error is in the zone boundary file and the rest of the error is related to link shapes.

The results of the point-in-polygon match within ArcGIS were applied to the activity location file using the TAZUpdate program. This program generates a list of zones that have zero or only one activity location. It is essential that all of the zones have at least two activity locations in order for ConvertTrips to allocate the intra-zonal trips. The TAZUpdate program identified nine zones with no activity locations and an additional nine zones with only one activity location. ArcView was used to review the zones that were reported as problems and manually update the zone assigned to selected activity locations. In most cases an activity location near the zone boundary could be re-assigned without significant distortion. A few zones contained no roadway access. In this situation it was necessary to identify where the local roadways entered the regional network and select an activity location in that vicinity.

3.3.2 Subzone Weights

Once the activity locations were assigned to the correct zone, a refined sub-zone allocation of trip ends was attempted. By default, the trip origins and destinations by trip purpose are evenly distributed to the activity locations within the zone. The user can restrict or refine the distribution by adding sub-zone weighting factors to the activity location file. METRO estimated these weighting factors using parcel data.

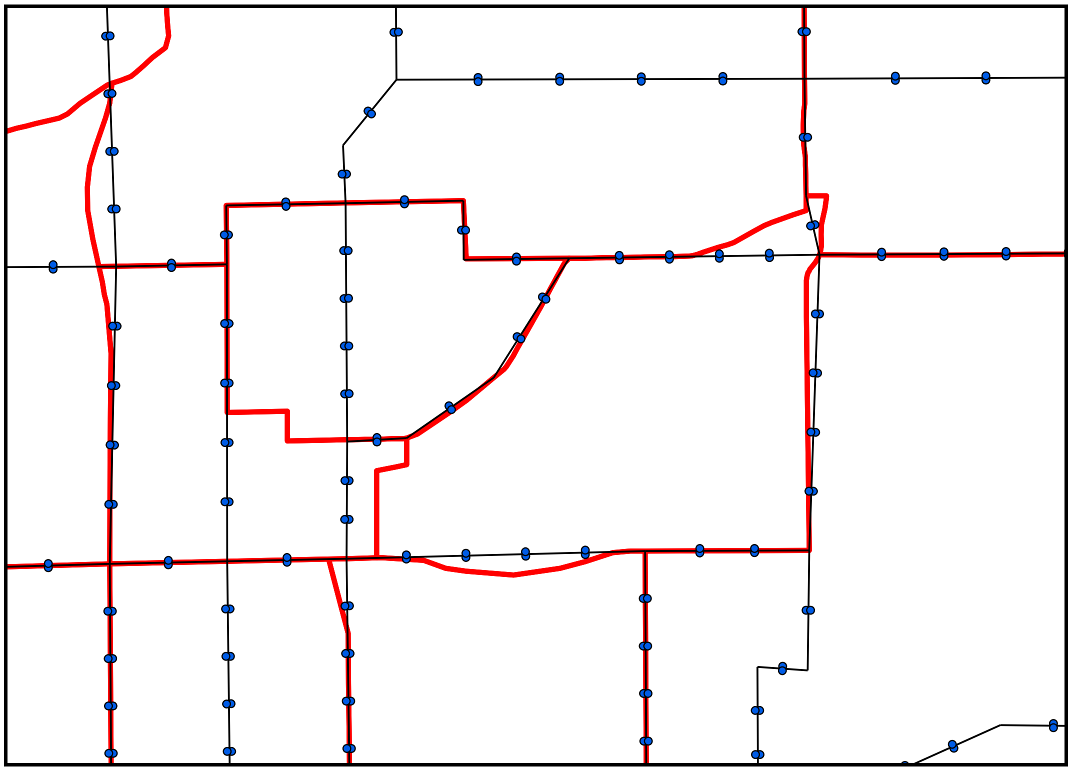


FIGURE 25: Activity Location Reassignment

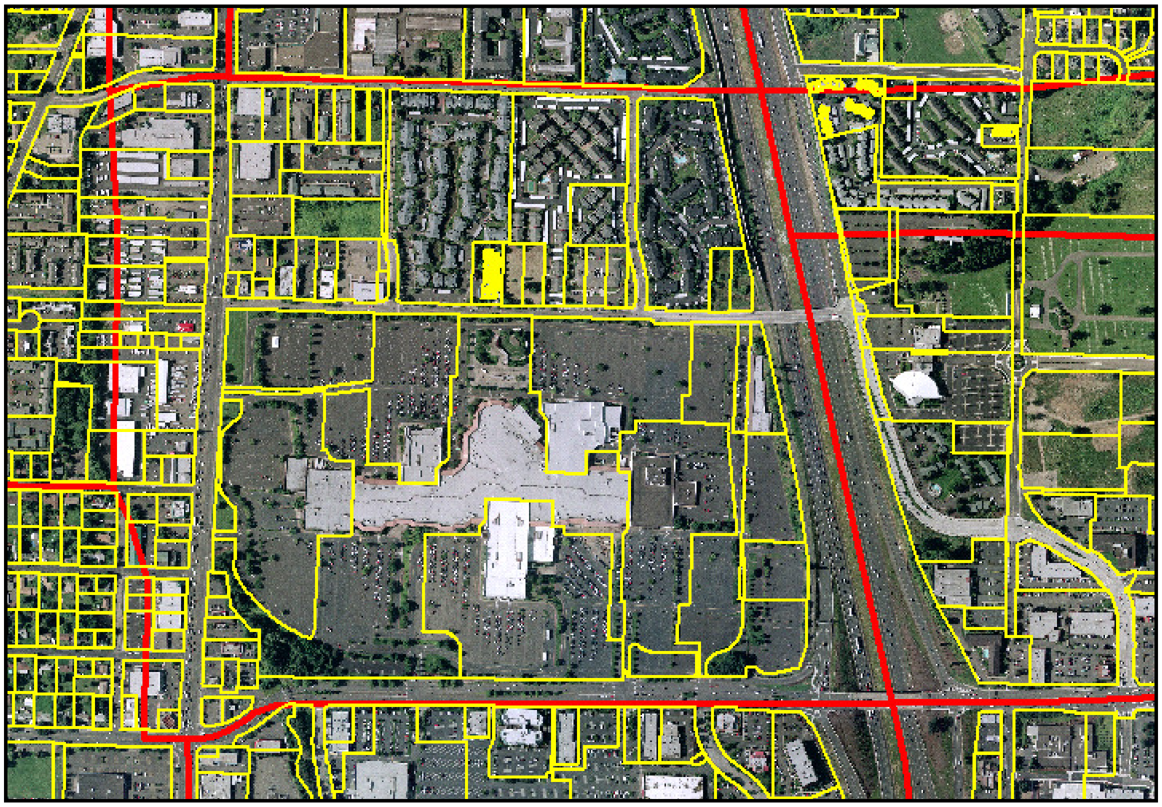


FIGURE 26: Parcel Polygon Overlay

Figure 26 overlays METRO's parcel data on top of an aerial photo with zone boundaries. METRO used ArcView to allocate the parcel data to zones and then assign each parcel to the nearest set of activity locations within the zone. If the parcel contained more than one activity location, the household and employment data associated with the parcel were equally distributed to the activity locations within the parcel. The user could then choose to use the household and employment data as sub-zone weights or apply trip generation rates by trip purpose to the household and employment data and store the trip productions and attractions by trip purpose in the activity location file.

3.3.3 Findings

The refinements discussed above improved the loading of traffic onto local roadways. This in turn, improved the distribution of traffic on the network and reduced the impacts of congestion on the overall performance of the simulation. This was particularly helpful near large shopping malls and major employers.

The refinements did not, however, have significant impacts on regional travel or screenline validation. If detailed parcel data or zone boundaries are not available, the default methods should work reasonably well in most situations. The user may wish to spend some time reviewing the allocation of trips around major traffic generators. If the Microsimulator shows some unexpected queues, one solution may be to review and refine the parking lot loadings near the congestion point. It may be possible to fix the problem through a more realistic allocation of trips.

Chapter 4 Modeling Tools

The TRANSIMS Router and Microsimulator are used to assign trips to the highway and transit networks. The network conversion process described in Chapter 2 was used to generate the synthetic network files required by TRANSIMS. The ConvertTrips program was then used to convert the daily trip tables from METRO's regional model into the activity data required by TRANSIMS. Given the TRANSIMS network and activity set, the trips can be routed through the network using the Router and assigned using the Microsimulator.

This chapter describes the overall TRANSIMS system architecture and how the software tools were used to develop modeling applications. It starts with model applications using the TRANSIMS Modeling Interface, the issues encountered, and the steps taken to improve the results. It then describes a number of techniques and tools that were developed to improve the process efficiency and minimize the total processing time for the Router-Microsimulator process.

4.1 TRANSIMS System Architecture

Figure 27 provides a high level view of the TRANSIMS system architecture. The Modeling Interface, Network Editor, and Output Visualizer are Windows-based applications that communicate with a Linux cluster to execute the TRANSIMS core technology. The Modeling Interface is a graphical user interface that allows the users to define configuration keys for various modules in the TRANSIMS core technology, and then run these programs, monitor the run status, and view the log files. It stores the configuration keys and an application log in a database for future reference and re-use. For this project, the TRANSIMS Modeling Interface was used to execute each step in the Router-Microsimulator process. The Output Visualizer was used to view the simulation results and identify network coding problems. The Network Editor was used to correct the network coding.

For major applications, the Linux cluster consists of 20 or more computers. In order to view, review and edit the TRANSIMS network files, the files are first imported into the Network Editor database. Network edits are made using the tools available within the Network Editor and the results are exported to the Linux file system. After the Microsimulator is run, the Output Visualizer is used to analyze the simulation results. The Output Visualizer reads the Linux data files into a custom animation database for display purposes.

File transfer between the Windows workstation and the Linux cluster is accomplished using FTP (File Transfer Protocol) or SSH (Secure Shell Protocol) utilities, or Linux terminal windows. In some cases, additional processing is required to correct the end-of-line syntax between the Windows and Linux operating systems. The "dos2asc" utility was developed to make these conversions. In all cases, some basic data manipulation and file processing needs to be done on the Linux system. Working knowledge of Linux commands and standard utilities (e.g., awk, vi, perl, etc.) is extremely helpful.

For more information about the Modeling Interface, Network Editor, or Output Visualizer, go to the IBM TRANSIMS Solution Center at http://www.transims.net.

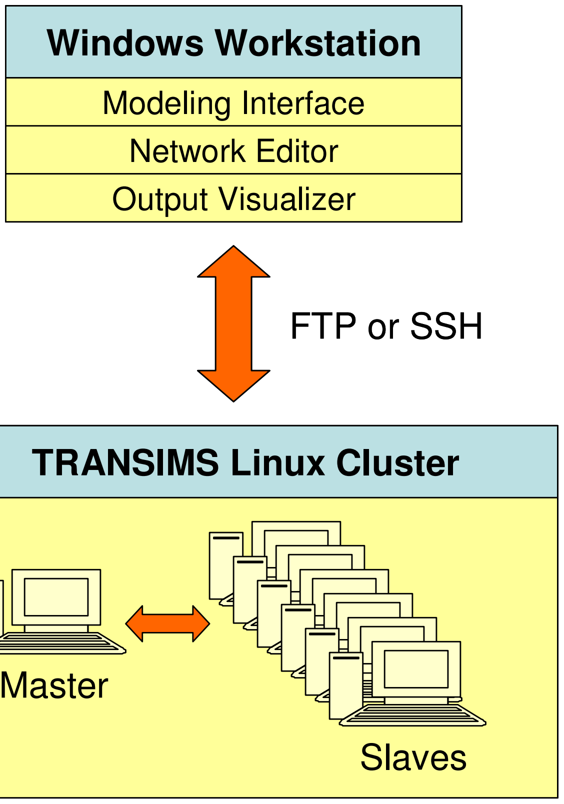


FIGURE 27: TRANSIMS System Architecture

4.1.1 Router-Microsimulator Steps

Figure 28 shows the sequence of steps that are typically needed to implement the TRANSIMS Router and Microsimulator using the Modeling Interface. The steps that are outside of the red rectangle identify tasks that need to be executed outside the Modeling Interface. These steps involve running PERL scripts from the Linux command line. Note that this discussion assumes that the TRANSIMS network and activity set were generated by the procedures discussed in Chapters 2 and 3. In other words, the network, activity, population, and vehicle files have all been created and indexed. If the files were copied from the Windows workstation or from one project directory to another on the Linux cluster, the corresponding index files will need to be recreated.

The Router requires a list of households that are to be routed for any given run. This could be a list of all of the households in the region or just the households that need to be re-routed due to congestion or some other issue. The household list needs to be generated by the user prior to entering the Modeling Interface. This file can be created in a number of ways, but it is generally best to start by extracting all of the Household IDs in the region from the TRANSIMS population file. A PERL script called "MakeHouseholdFile" is provided in the TRANSIMS software directory for this purpose. This script needs to be executed outside of the Modeling Interface.

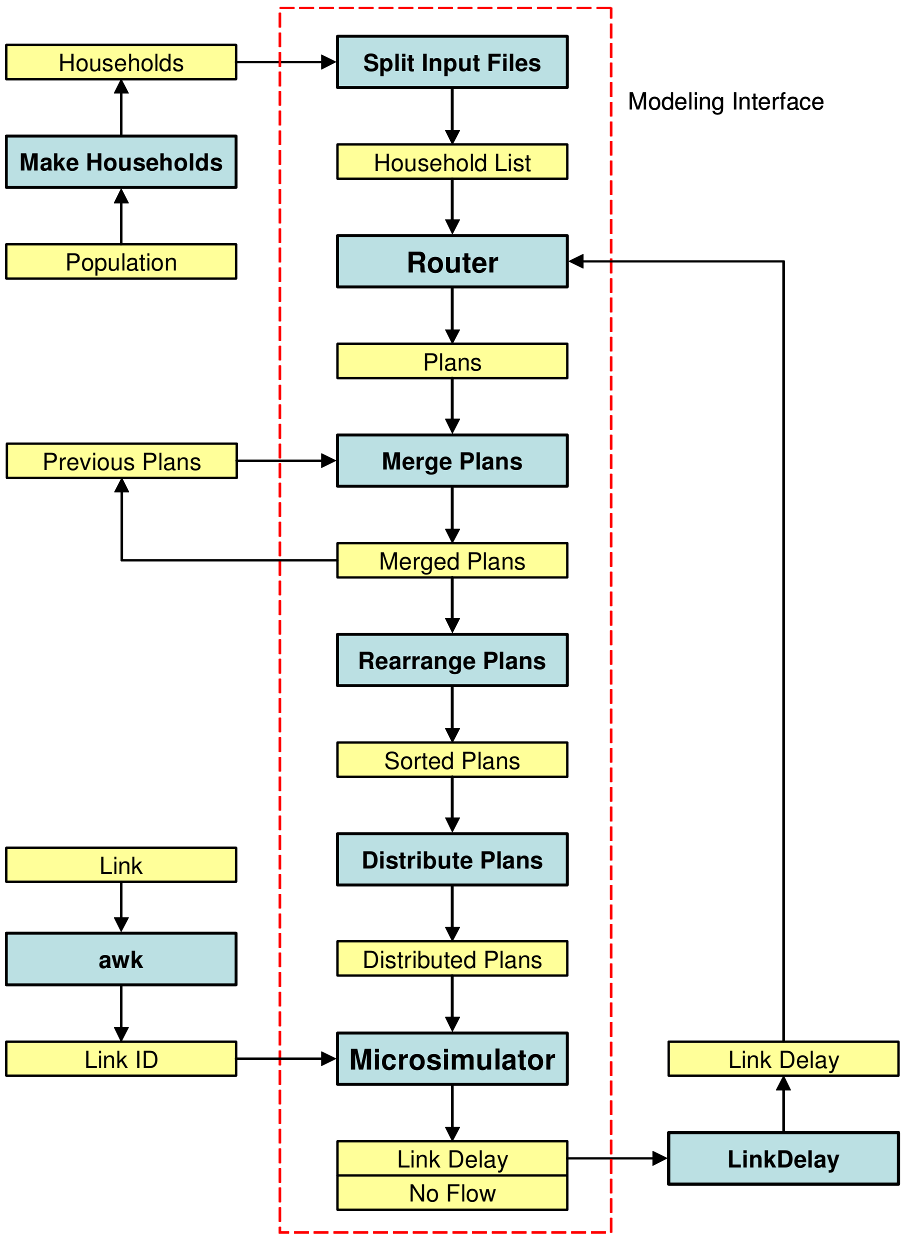


FIGURE 28: Router-Microsimulator Steps

The user will also typically create a list of Link IDs prior to executing the Modeling Interface. The Link IDs are needed by the Microsimulator to generate output data files. These files include the link delays and snapshot information typically used to visualize system performance and calculate travel time. If the user is familiar with the “awk” utility, a simple “awk” command can be executed to extract the Link IDs from the network Link file. This process should be re-run any time links are added to the network.

Once the preliminary data processing steps are completed, the Modeling Interface can be used to execute the Router-Microsimulator process. This process begins by splitting the household list into separate files for each computer in the Linux cluster. The Modeling Interface task labeled “Split Input Files for Parallel

Execution" is provided for this purpose. The user specifies the number of partitions and the procedure randomly assigns households to each partition.

The user then sets the Router configuration keys to identify the appropriate input and output file names and processing options and starts the execution of the Router using the "Run Router" interface. The Router generates travel plans for each person in each household. The plans include the time and duration of each activity and the travel path between activities. The paths are mode specific and identify each leg of the trip in detail (e.g., walk to the vehicle, drive on specified links to a parking lot, and walk from the parking lot to the activity location).

If this is the first time the process is executed, the merge plans step is skipped. Otherwise, the newly created plans need to be merged with previously generated plans using the "Merge Base and Partial Plan Files" procedure. This procedure replaces existing plans that have been re-routed and adds any new plans to create the complete plan set that is to be simulated.

The "Rearrange Plans" interface is then applied to validate and sort the plans by time of day for input to the Microsimulator. The new plan index files are then distributed to network partitions using the "Distribute Plans" interface. The distributed plans are then simulated using the "Run Microsimulator" module. Configuration keys are set within the "Microsimulator" module to generate a variety of output files from the microsimulation. The most significant output files generated from the process include the snapshot and link delay files.

In current practice, two link delay files are used to summarize the total travel time and volume on each link in 15-minute periods throughout the day. One file summarizes the performance based on the number of vehicles that leave the link during the 15-minute period. The other file records information for links where no vehicles leave the link during the 15-minute period. This could be low volume condition for which the free flow speed would best represent the travel time on the link, or it could be a total gridlock situation where traffic doesn't move at all. In order to append the records of the "no flow' file to the "link delay" file, the "LinkDelay" PERL script was run from the Linux command line. The LinkDelay script uses a very slow speed (0.5 meters per second) to estimate the travel time on congested links for use in subsequent path building. After the link delay file is updated using the "LinkDelay" utility, it is used as an input to the Router for the next phase of path building.

4.2 TRANSIMS Assignment Procedures

The Router-Microsimulator steps described above represent one iteration of the process that ultimately identifies travel paths for all travelers. The Router identifies the minimum time path during each iteration based on the current set of travel times. The Microsimulator loads these paths to the network and calculates the resulting travel time on each link by time of day. If the new travel times are different from the ones used by the Router, trips are re-routed to adjust the travel paths. The process continues until the stopping criteria are reached.

This iterative feedback process represents the TRANSIMS assignment procedure. It is completely flexible and can be implemented in a variety of ways to achieve a wide range of objective functions. The Portland implementation project tested a number of approaches. All of these approaches attempted to achieve the fundamental objective of minimizing the travel time for each traveler. The approaches differed in the methods used to achieve this objective. The primary evaluation measure for a given method was how long it took to converge on a reasonable solution.

4.2.1 Simple Assignment Process

In this approach, all of the trips are routed using free flow speeds and then the resulting travel plans are simulated using the Microsimulator. The link delays from the Microsimulator are used to re-route a random sample of household. The new routes replace the original paths and the Microsimulator is re-run. This iterative feedback process continues until the simulation results are reasonable.

The major disadvantage of routing all of the trips using free flow speeds is that it tends to generate more demand than can be accommodated on critical links in the network. When the network is congested, the Microsimulator created cascading queues that cause severe gridlock. This significantly deteriorates the performance of the Microsimulator and results in unreasonable link delays for re-routing trips during feedback loops. A large number of feedback iterations are required to correct these inaccuracies.

4.2.2 Incremental Assignment Process

To reduce the number of iterations required to generate a reasonable distribution of traffic, an incremental assignment method was tested. The total households were randomly split into seven groups containing 30, 20, 20, 10, 10, 5 and 5 percent of the households. The Router was run using free flow speeds to generate travel plans for the first 30 percent increment. The trips were simulated and the link delays generated by the Microsimulator were used to route the second 20 percent increment. The 20 percent plans were then merged with the first 30 percent plans to create the 50 percent plan set which was simulated to generate link delays for routing the next increment. The objective was to continue the process until all of the increments were loaded.

The Microsimulator had no difficulty loading 70 percent of the trips, but when the next 10 percent increment was added, the Microsimulator produced severe gridlock and cascading queues that overwhelmed the Microsimulator. A complete simulation was never achieved when loading 80 percent of the trips. About 15 percent of the trips were lost (i.e., off plan) due to network congestion issues, and a significant number of vehicles had not completed their trip at the end of the simulation period (3:00 AM). A small percent of trips are expected to be traveling on the network at the end of the day, but in this case most of the vehicles were stuck in queues that never dissipated. The Output Visualizer shows these types of problems very clearly.

4.2.3 Incremental Feedback Procedures

The incremental assignment results suggested that some re-routing was necessary to keep the Router from overloading key bottleneck locations. The RandomSelect program was written to randomly select a percent of households from the original household partitions for re-routing. This program selects a user-defined percent of records from any number of files. The program was used to select 20 percent from the 30, 20, 20, and 10 percent increment files for re-routing using the 70 percent loaded speeds generated by the Microsimulator. Figure 29 shows how the households were selected for re-routing.

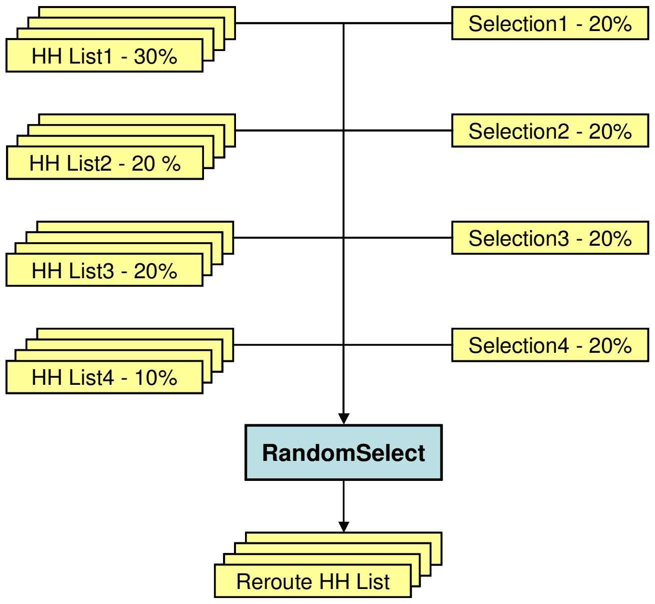


FIGURE 29: Random Re-Routing

The random feedback was integrated into each incremental loading step so that a small percentage of trips were re-routed as each new increment of trips was loaded onto the network. For example after the first 30 percent trips were simulated, 20 percent of the 30 percent trips were selected for re-routing using the link delays generated by the 30 percent assignment. These plans were added to the next 20 percent increment and re-simulated. This process allows a portion of the travelers to reconsider their path after each step of the loading process. This helps address congestion problems before they create cascading queues and severe gridlock. The random re-routes made it possible to simulate all of the trips with realistic congestion levels.

While the results were encouraging, the initial processing time required to achieve the results was disturbing. The Microsimulation runs were taking eight to ten hours to complete when the network was congested. Attempts were made to increase the process efficiency so that the results can be produced and reviewed more quickly. The sections below describe the steps that were taken to improve the process efficiency.

4.2.4 Time Based Loading Method

One the reasons why the incremental approaches described above took considerable time to run was because it involved a complete simulation of the whole day as part of each iteration. When the network is congested, cascading queues form that significantly reduce the performance of the Microsimulator and make the resulting travel times illogical. One way to avoid this problem is to address congestion issues early in the simulation before the vehicles have a chance to collide with other congestion problems and create gridlock. This was the basic objective of the time-based loading method.

The time-based approach routes and simulates all of the trips up to a given time of day. The time of day was selected based on the network congestion (i.e., the Microsimulator was terminated when cascading queues started to form). The method then re-routes a percentage of these trips until the simulation settles down and generates reasonable results. At this point, trips in the next time increment are routed and simulated. The process continues until all the trips are simulated.

The only real complication with this method is identifying which households make trips at a given time of day. Since the Router operates on a list of households and all trips for all persons in the household are routed as a group, it is not as straightforward as might initially be expected to assign households to time periods. Fortunately, this TRANSIMS implementation made each trip in the Portland METRO trip tables into a unique household-person record. This made it possible to select households based on the start time of the single trip the household made during the course of the day.

4.2.5 Time Based Feedback Procedure

A further refinement of the time-based loading procedure included incremental feedback. Rather than clearing all of the cascading queues during a given time period before moving on to the next time period, this approach removed only the major problems in each time period. Incremental feedback was used to address the less significant problems as part of subsequent runs. This method takes advantage of the fact that the Microsimulator needs to be run from the beginning of the day (i.e., midnight) each time it is executed. As each subsequent time period is added to the analysis, all of the previous time periods also need to be simulated. The process continues to re-route a portion of trips in the earlier time periods to refine the travel times and address minor problems.

Table 2 shows an example of how this method was implemented at METRO. METRO divided the day into five basic time periods (midnight to 9 AM, 9AM to noon, noon to 3 PM, 3 PM to 6 PM, and 6 PM to midnight). To begin the process all of the trips starting before 9:00 AM were routed using free flow speeds and simulated until noon. Twenty percent of the travelers who started their trips before 9:00 AM were then randomly selected and re-routed using the Microsimulator travel times. The process continued until the simulation produced reasonable results through 9:00 AM (in this case three feedback iterations were used). The trips that started between 9:00 AM and noon were routed using the latest Microsimulator travel times. The next feedback iteration re-routed 20 percent of these trips plus 10 percent of the trips that started between midnight and 9:00 AM. The process continued until all five time increments were simulated and the congestion problems are addressed.

A small percentage of the earlier time periods are always re-routed during the later iterations. When the process is complete, trips during the early hours of the day are reasonably stable, but the trips at the end of the day could still be improved. Additional feedback iterations that focus on re-routing PM Peak and evening trips are needed to stabilize the whole system.

TABLE 2: Time Based Feedback Procedure

Step	0 to 9	9 to 12	12 to 15	15 to 18	18 to 24	MSim
1	20%					12:00 PM
2	20%					12:00 PM
3	20%					12:00 PM
4	10%	20%				3:00 PM
5	5%	15%	20%			6:00 PM
6	5%	10%	15%			6:00 PM
7	5%	5%	10%	20%		9:00 PM
8	5%	5%	5%	20%		9:00 PM
9	5%	5%	5%	20%		2:00 AM
10	5%	5%	5%	15%	20%	2:00 AM
11	5%	5%	5%	10%	20%	2:00 AM
12	5%	5%	5%	5%	10%	2:00 AM

4.2.6 Process Efficiency Issues

All of the assignment procedures outlined above were implemented using the TRANSIMS Modeling Interface. One limitation of the Modeling Interface is that it does not allow the user to run a sequence of steps at one time. The user executes a program, waits for the program to finish, modifies the configuration keys for the next program, and executes that program. Since the execution of the Router-Microsimulator process required continuous user intervention, this was inefficient because the user was not always available to start the next process at the completion of the previous step (e.g., 3:00 AM).

This is further compounded by the need to run a few steps of the process outside the Modeling Interface. For example, in order to execute the incremental feedback process, the RandomSelect program was run outside the Modeling Interface to select the households for re-routing.

In addition, setting up the model runs using the Modeling Interface was not a trivial task. A number of configuration keys had to be reset between successive feedback runs. The fact that these keys are set at different places within the Modeling Interface makes it susceptible to user error. For example, if the user forgets to reset the Link Delay key or types the file name incorrectly, the Router generates paths for travelers using the wrong travel times. Careful attention to the file names was also required so that successive feedback runs could be executed with minimum confusion by multiple staff members.

An automated process that executed the runs with minimal user intervention and user error was developed. One quick fix solution was to create configuration files for each step of the incremental process using the Modeling Interface and then use these files in a simple batch process using PERL scripts. In order to make the process work, a few standard keys defined in the Modeling Interface were used to pass parameters to the batch process. A batch command line file was then created and executed to run a specific increment of the assignment process. The process was applied to a new alternative or increment by copying and editing the configuration files so that the keys point to the appropriate file names. This approach significantly reduced the time required to setup and executes multi-step runs.

4.3 Batch Processing

The evolution of the TRANSIMS assignment process ultimately lead to the formal development of a batch processing system. The RunSetup program was developed to automate the whole process of setting up the configuration and batch files needed for a given run. Standard naming conventions with key word replacements were introduced to make the model implementation easier and more dependable. Most of the data process tasks and report generation routines that were originally executed outside of the Modeling Interface could now be integrated into the automated process.

The batch system greatly reduced the user error and improved the process efficiency. It also helped to monitor the process status. In order to provide the user with application flexibility, the batch process was divided into three parts. Each part can be run independently or chained together into a series of model runs. The three parts are called

1. Router Process
2. Time Sort Process
3. Msim Process

The Router Process is used to build travel plans for selected travelers, and merge the new plans with the previous travel plans to create a set of updated travel plan files. The Router Process also generates a number of summary reports and data files based on the updated plans. The Time Sort Process sorts the plans by time of day and combines the sorted plans into the index files required by the Microsimulator. The Msim Process simulates the travel plans and summarizes the outputs. The Msim Process includes a number of utilities to analysis the Microsimulator results and document the system performance.

When the three steps are run together, they represent a complete Router-Microsimulator iteration. A feedback process based on multiple iterations is typically required to generate realistic results. The batch processing tools include standard naming conventions which allow the user to combine multiple feedback runs into an over-night or weekend process. The batch process generates log files that are used to check the status of each run. This section describes each of the processing steps in detail. The next chapter describes how the steps were combined into complex assignment models.

4.3.1 Router Process

The first task in the Router-Microsimulator process is to build paths through the network from one activity location to another based on the current link travel times and time of day constraints for the specified mode of travel. The Router builds the point-to-point paths and outputs the detailed travel plans for selected travelers. It is important to understand that the Router builds a unique plan for each traveler based on the sequence of activities the person is engaged in over the course of the day. The link times used to construct the path are based on the average 15 minute travel time on each link at the time of day when the traveler is expected to enter the link. The link times also include random variation to simulate perception differences among travelers. If the trip is made by transit, the route service and waiting time are based on the time of day when a specific transit vehicle is scheduled to leave the transit stop and the time of day the traveler arrives at that stop.

The other important point is that plans are only developed for selected travelers. The Router is provided with a list of households and it generates a set of plans for each traveler within each household. This is done for two reasons. The first is to facilitate partitioning – the division of effort between Routers running on multiple computers. The second is to facilitate feedback – updating the plans for specific travelers.

Figure 30 shows the components of the Router Process. The figure includes three columns of Router and PlanPrep processing to depict the multiple partitions typically included in a TRANSIMS application. The TRANSIMS architecture allows splitting the input household files into multiple partitions so that the routing, plan processing and microsimulation can be run on multiple nodes or linux machines to increase the process efficiency. A large application can easily include 25 or more partitions. Each partition runs independently on a separate computer. The PlanSum program combines data from all of the partitions to generate the composite performance of the system.

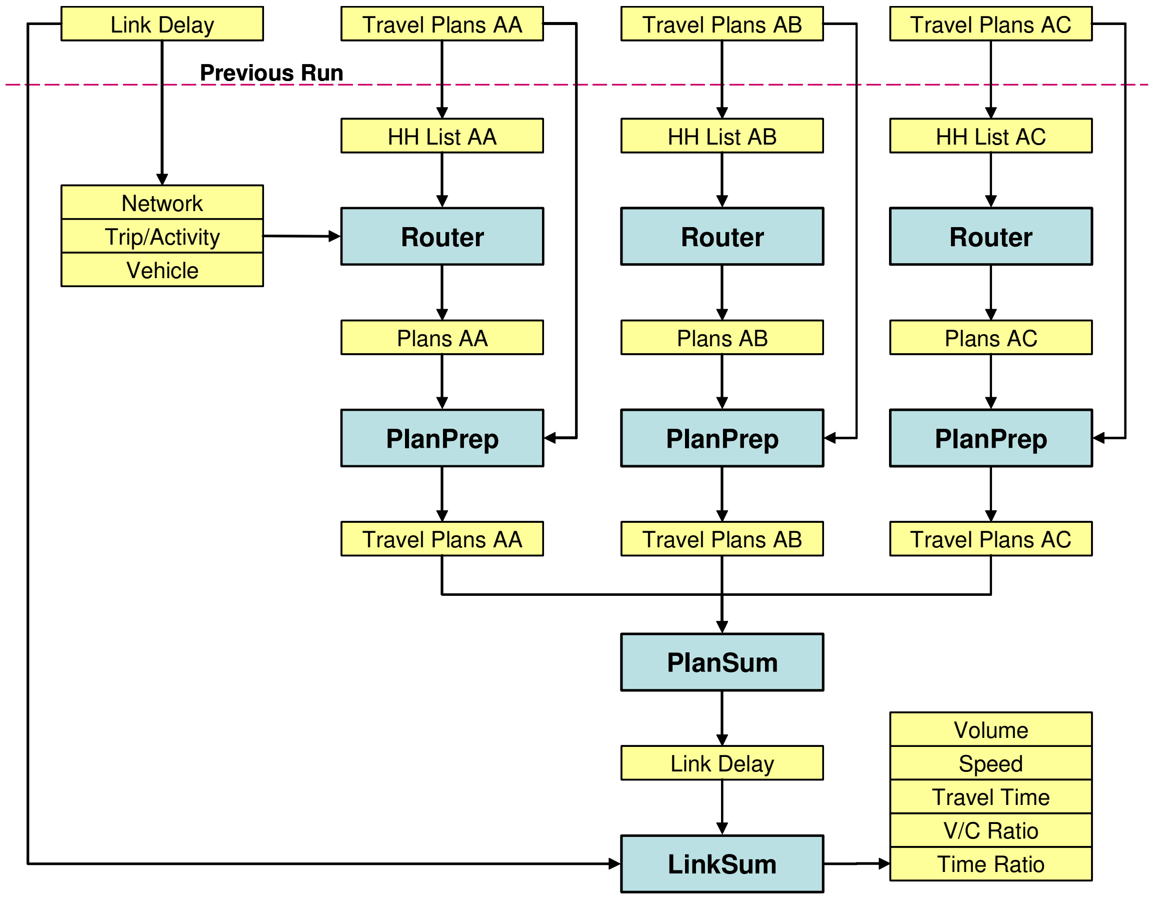


FIGURE 30: Router Process

The Router Process in essence replaces the "Run Router" and the "Merge Plans" modules that exist in the TRANSIMS Modeling Interface. The household list files tell each Router application which households to process. Each Router generates plans for those households using the link delays generated from the previous run. PlanPrep is then used to sort and merge plan files by traveler. If the new plan file includes a traveler that was previously routed, the new plans will replace the previous plans for this traveler. It also generates the traveler or time index files needed by other TRANSIMS programs.

The plan files generated by the Router are huge. For example, in the Portland application the size of each plan file is typically 350 megabytes and it is difficult to view or open the files using standard text editors.

The PlanSum program reads the plan file from each partition and generate summary files and performance reports. This includes link demand summaries by time of day, volume-capacity ratio reports, and link travel times for each 15 minute increment based on volume-delay relationships. The link delay file can be read by the LinkSum program to generate a number of link-based statistics.

4.3.2 Time Sort Process

The plan file generated by the Router contains a list of all of the travel activities planned for a given traveler over the course of the day. The Microsimulator expects the activities for all travelers to be sorted by the time of day when the activity begins. The Time Sort Process uses the PlanPrep program to sort the traveler plans by time of day and merges the plans from each partition into a single plan index for input to the Microsimulator. The Time Sort Process produces a single plan index file that is sorted by time. The Microsimulator requires indices that are sorted by traveler ID in addition to the indices that are sorted by time. The FlipIndex program reads the time sorted indices and generates an index file that is sorted by traveler ID. Figure 31 depicts this process.

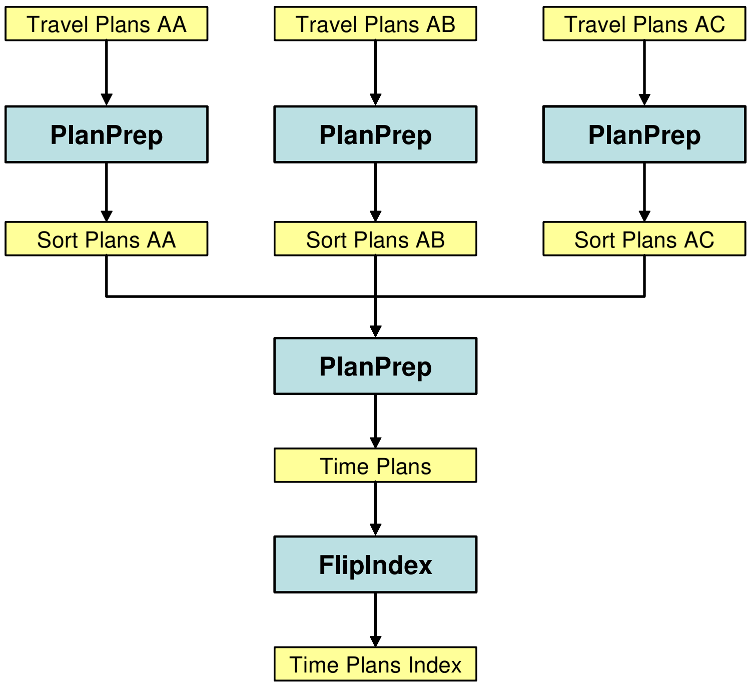


FIGURE 31: Time Sort Process

4.3.3 Msim Process

The Microsimulator is significantly different from the Router in the way it implements partitioning. The Router runs as an independent application on each partition. The Microsimulator physically partitions the network and runs a "slave" application to process the vehicles within each partition. At the end of each time step (one second), the slaves send messages back to the master node about vehicles that need to be moved from one partition to another. The master node coordinates these transfers and keeps all of the slaves synchronized.

What this means from the point of view of the batch process is that the Microsimulator behaves like a stand-alone application. This is why the Microsimulator is shown as a single program in the Figure 32. It generates one set of files from the master application. The set of output files can vary considerably. The typical setup used for this study includes a log file, two link delay files, three event files, and one or more snapshot files. The Msim Process is used to run the Microsimulator and summarize the output files to generate summary results and reports. The reports and summary files help the user quickly review the simulation results.

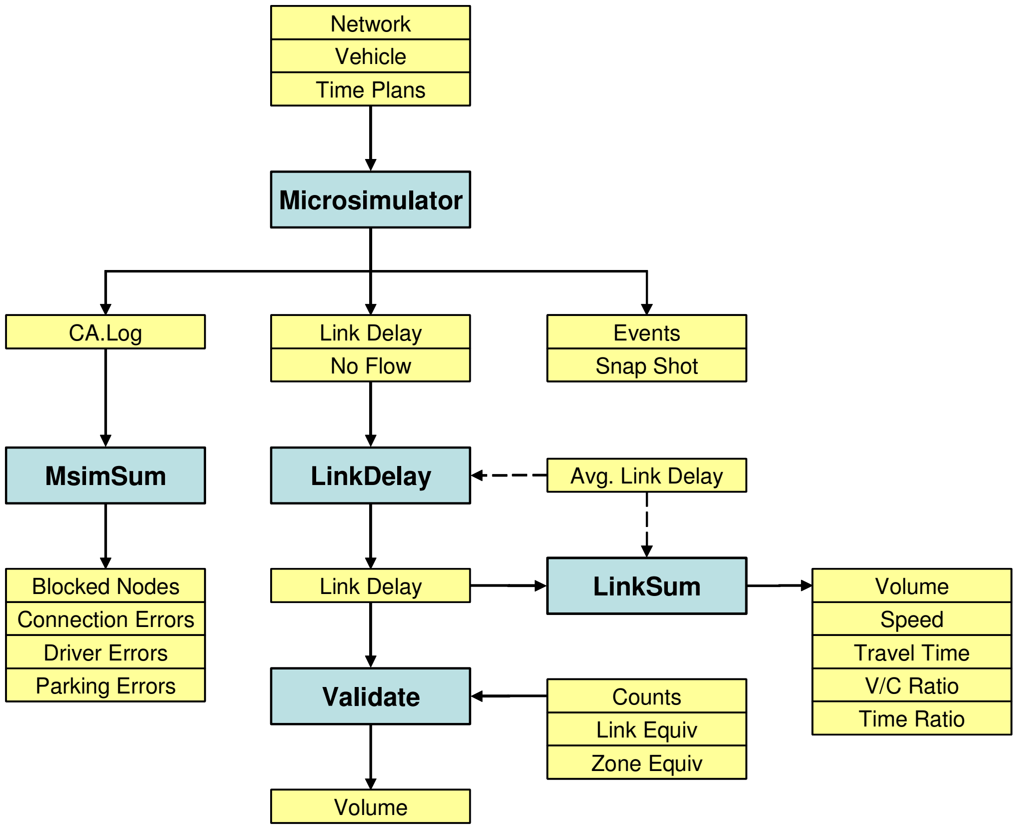


FIGURE 32: Msim Process

The event and snapshot files generated by the Microsimulator provide information that can be summarized in reports or displayed in the Output Visualizer. Reviewing the snapshot data in the Output Visualizer is often the most effective and efficient way to interpret the performance of the network and identify coding errors or issues that need to be addressed. It can also provide a qualitative assessment of the overall congestion levels in the system. It does not, however, help with convergence or stopping criteria. This determination requires the quantitative data that the link delay file provides.

The Event files are used to output the events each time an event of specific interest occurs. The Event files that are generally produced are the trip start times, trip end times and lost vehicles or off-plan event files. The start time event file records the time when a vehicle is loaded onto the network. Similarly, the end time event file identifies the time when the trip is completed. The off-plan event file stores information about the vehicles that go off-plan during a trip. The vehicles go off-plan if they are not in an appropriate lane to complete the required maneuver when they enter an intersection. The vehicles also go off-plan if they get stuck in a queue without moving for a user-specified amount of time. The Portland study experimented with a number of maximum time values. Most simulations were done with a maximum delay of ten minutes.

The LinkDelay program in Figure 34 replaced the “LinkDelay” PERL script provided by LANL to append records from the “no flow” file to the link delay file. This LinkDelay programs has the capability of reading two link delay files and producing either a simple averaged or a weighted averaged link delay file. The resulting link delay file becomes the centerpiece of a number of performance analysis routines. Link delay averaging is useful in reducing the oscillating effects between successive feedback runs. Typically the link delay averaging is used only during the later stages of stabilization runs.

The LinkSum program summarizes the link delay file and produces information in a number of ways. It generates cumulative distribution reports of various performance statistics by time of day, functional class, and link group. It also generates volume, speed, travel time, V/C ratio, and time ratio data files that can be joined with the network link file to display or plot traditional statistics on maps. The LinkSum program can also be used to compare the performance of one run to another to determine how much the system performance changed on a link-by-link basis. Stable link results are one way of quantifying convergence. It should be noted that the link delays generated by simulating the same set of trips twice can yield slightly different travel times due to the stochastic nature of the Microsimulator.

The Validate program is executed by the Msim Process to compare the simulation results to observed traffic counts. The results from links that include traffic counts are statistically compared to the counts and summarized in a variety of ways. Summaries are typically generated by volume group, functional class, area type, and link group or screenline. Statistics include the percent and absolute difference, the percent and absolute average error, the standard deviation, the correlation coefficient and the percent root mean square error. This tool can also be used to compare one alternative to another or the TRANSIMS volumes to the traditional model results.

Chapter 5 Stabilization Process

The application experiments and modeling tools described in Chapter 4 became the basic building blocks for the assignment and stabilization process that was ultimately developed by this study. Chapter 4 discussed a number of assignment techniques that were tested using the TRANSIMS modeling tools. The incremental assignment procedures generated promising results, but there was no real effort to define quantitative stopping criteria. The process was basically run until the Output Visualizer showed the expected levels of congestion by time of day and the assigned volumes reasonably matched the traffic counts. There was no proof that the travel times were optimal for any given traveler or that the network statistics were stable. In order to be useful, it was necessary to develop a process that approximated user-equilibrium within the TRANSIMS framework using quantitative stopping criteria.

This chapter describes the loading and stabilization process that was developed for the Portland study. It is by no means the only way TRANSIMS could be employed. It may not be the best way. Additional experience is likely to identify a number of improvements to the process as it evolves over time.

5.1 TRANSIMS Considerations

A traditional equilibrium assignment model basically runs the Router with a function at the end that determines the best way to combine the current results (i.e., volumes) with the previous results to minimize travel time. When the contribution (i.e., fraction) of the current run to the combined result becomes small, the process stops and equilibrium is declared. The TRANSIMS trip assignment process attempts to achieve the same general objective in a different way. First of all TRANSIMS cannot deal with fractions of trips on links. It needs to model individual people making complete trips. So rather than combining trip fractions from a given iteration, TRANSIMS selects a fraction of the travelers to combine or re-plan. When the number of travelers who can improve their travel time by changing travel plans becomes small, the process terminates.

In addition, the fact that TRANSIMS simulates individual people and vehicles makes the assignment dynamics significantly different from a traditional model. A traditional model adds a vehicle to the volume total on every link along the minimum time path regardless of the traffic or travel time on the route. It does not constrain the assignment based on the capacity of the roadway. It attempts to restrain the loading by making the travel times on over-capacity links large. TRANSIMS does constrain the assignment to the physical number of vehicles that can be accommodated at a specific time of day. If a traveler runs into a capacity constraint, the remainder of the travel plan is either delayed or lost. If the bottleneck is cleared, congestion problems can easily move to downstream links. This makes it relatively important to address bottleneck locations as soon as possible in the assignment process.

The final consideration is the processing time. A traditional model re-routes all of the trips during each iteration. Since the processing time to route and simulate a full set of travel plans within TRANSIMS is significant, it is important to avoid wasting time on travelers who don't need to be reconsidered or shouldn't be reconsidered until compounding issues are addressed. The goal is to design a process that focuses the computational resources on the most significant problems first before it looks at subtle changes in other areas. For example, there is very little reason to continue running the Microsimulator once the network reaches total gridlock. It is better to stop the process and address the source of the problem earlier in the day before proceeding.

Given these considerations and based on the experience gained while running various assignment methods, a multi-dimensional process was developed to achieve a user-equilibrium result within the TRANSIMS framework. This process begins with an incremental loading technique using volume-delay equations to initialize the plans and then proceeds to a series of stabilization and convergence steps to refine the plans until the travel times are minimized for each traveler.

5.2 Initializing the Travel Plans

The assignment experiments demonstrated that a large number of Router-Microsimulator iterations were needed to produce a reasonable result. The total processing time required for a complete assignment was significant. Though the batch procedure makes the process efficient and quicker to run, it does not reduce the computer processing time. To achieve this goal, the process needed to provide the Microsimulator with a more realistic set of travel plans. Running the Microsimulator to calculate the link travel times for a set of illogical plans was time consuming and in some cases counterproductive. What was needed was a better estimate of the link travel times by time of day so the Router could develop travel plans that the Microsimulator had some hope of simulating.

5.2.1 Router Enhancements

A research effort at Virginia Tech tested enhancements to the TRANSIMS Router that updated the travel times on the links after each traveler was loaded. Volume-to-capacity relationships were used to estimate the link travel times for each time period. This provided the Router with better estimates of the link travel time by time of day which in turn generated a more realistic distribution of paths prior to the initial simulation.

For these and other performance issues with the Router, a new Router program that incorporated the V/C delay methodology was developed. The new Router used a BPR formula and link capacities to update the 15 minute travel times on each link at user-defined intervals. Since the new Router was also faster and addressed several routing issue in the original code, it was decided to replace the original TRANSIMS Router with this new Router to make the process more computationally efficient. The new Router does not do all of the household coordination tasks of the TRANSIMS Router, but it does serve the needs of this study quite nicely.

5.2.2 Initialization Procedure

Given the capabilities of the new Router, the trip assignment process used an initialization step based on traditional V/C ratios and volume-delay equations to estimate link travel times by time of day and generate a reasonable distribution of trips on the network. An incremental loading method was selected for this initialization because it focuses on individual travelers and it only routes each traveler once (i.e., minimizes processing time). Frequent travel time updates were included to reduce the likelihood of overloading links.

The process starts with a list of households and travelers within the region. The HHLists program extracts the household ID's from the input population file and randomly distributes the households to CPU partitions. The households are randomly allocated to partitions for parallel processing on multiple CPUs. The households in each partition are then randomly split into percentage increments using the Increments program. For the Portland study, the household file was split into sixteen increments. Each of first four increments included ten percent of the households and the last 12 increments included five percent of the households. Figure 33 shows the process used to generate the household increments for three partitions.

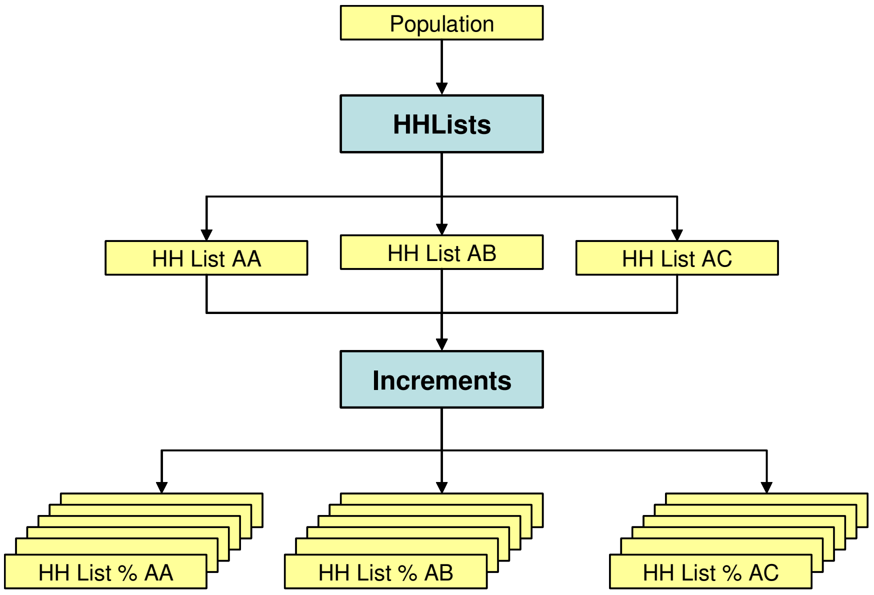


FIGURE 33: Creating Household Increments

After the Household increment files are created, the Router Process is used to execute the incremental loading process. The Router reads the input household list and generates a set of plans for each traveler within each household. The first ten percent of the trips are routed starting from free flow speeds. As the Router generates paths for travelers, it increments the 15-minute volume on each link the path travels through based on the time of day when the trip is expected to enter the link. The travel times on each link were updated after 1000 trips were loaded onto the network. A standard BPR formula was used to calculate the travel time using the 15-minute volumes and capacities.

Since each Router runs independently on each partition, the V/C-based travel time updates are made independently for each Router partition. In other words, partition one does not have access to the volumes loaded by partition two when calculating the travel time used for subsequent path building. The PlanSum program was developed to address this issue. After all the trips in the first 10 percent increment are loaded, the PlanSum program reads all of the plan files, traces the paths, and sums the volumes on each link in 15-minute increments. After all of the plans are processed, it uses the BPR formula to calculate the 15-minute link travel times and outputs a link delay file.

The link delay file generated by PlanSum was then used by the Router to initialize the volumes and travel times for each 15-minute period on each link. These travel times were used to build paths for the second 10 percent increment. This Router application also updates the travel times on the links as trips are loaded. The new volumes loaded by each Router partition are added to the total volumes from the previous increment to approximate the new travel time.

The plans from the second 10 percent increment are then merged with the first 10 percent increment to create a 20 percent plan set. The PlanSum program reads the 20 percent plan set to update the cumulative link delay file. This information becomes the basis for building paths for the next increment. The process continues until 100 percent of the trips are loaded onto the network. By splitting the process into 16 increments, the inaccuracies caused by partitioning are minimized. This enables the process to generate a reasonable set of initial plans. Figure 34 shows the overall process. The whole initialization process was executed on the METRO cluster in 2.5 hours.

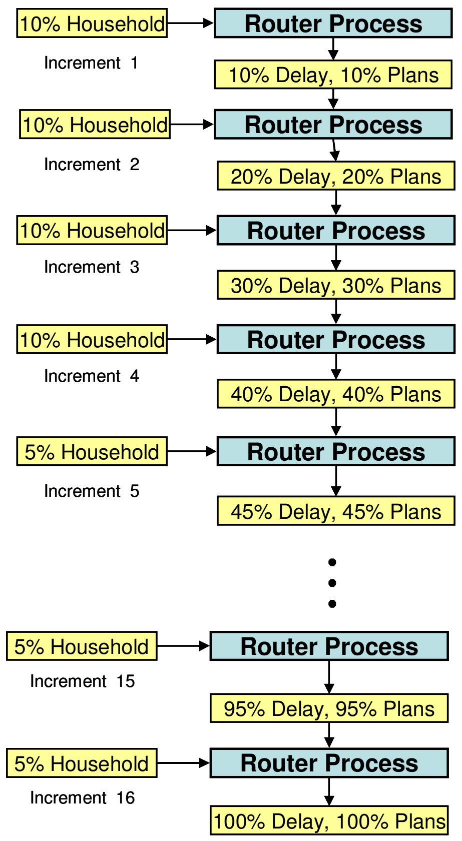


FIGURE 34: Incremental Loading Process

5.2.3 Random Feedback Iterations

In the incremental loading process, the minimum travel time path for each traveler is constructed based on the travel times on the network at the point when the plan is built. Travelers who are loaded in the first step of the loading process use free flow speeds and travelers who are loaded at the end of the process use loaded speeds. Since travelers are not allowed to reconsider their path, the initialization process creates situations where the travel time used to build the path is significantly different from the travel times computed after loading all of the trips.

To address this issue, several feedback iterations were added to the end of the incremental loading process to give the Router a chance to adjust the travel plans based on better travel time information. A random percent of households were selected for re-routing. These plans replaced the original plans in the plan set and the link travel times were updated based on the adjusted link volumes. The percentage of trips selected for re-routing in any given iteration was kept relatively small (less than 10%) to avoid major oscillation effects that can occur when large percentages of trips are re-routed. The selection criteria were also more heavily weighted toward households that were assigned to earlier increments in the initialization process. The RandomSelect program described in section 4.3.3 was used to construct the list of households for each iteration.

5.3 Stabilization Process

The initial plan set generated by the incremental loading process is a reasonably good starting point for running the Microsimulator. The Microsimulator will still generate some cascading queues during the peak periods, but most of these queues dissipate before they collide with other queues and cause total gridlock. The user could start running Microsimulator feedback iterations at this point and eventually achieve a stable solution. If the objective is to generate a set of travel paths were each traveler cannot change paths and reduce the travel time for their trip (i.e., user equilibrium), a stable simulation alone may not generate the desired outcome. Since the goal of this project was to produce a stable simulation and approximate user equilibrium at the same time, a stabilization process was designed to simultaneously migrate toward both objectives.

For computational efficiency, the stabilization process was implemented in five stages. The first two stages used the Router to improve the allocation of trips to network links and address travel time discrepancies for each traveler. Router stabilization was then followed by three types of Microsimulator stabilization. The first focused on resolving bottleneck locations within the network. The second stage focused on stabilizing the link travel times. A brief description of each of these stages is listed below. The following sections discuss each procedure in more detail.

1. V/C Ratio Stabilization – This is a Router based stabilization process in which the travelers who pass through links that have a high volume-to-capacity ratio are selected and re-routed.

2. Travel Time (Router) Stabilization – This is a Router based stabilization in which travelers who experience a significant difference between their expected travel time and the travel times computed from link delays (generated by PlanSum program using the plans) are selected and re-routed.

3. Random Stabilization Process – This is a Microsimulator based stabilization process in which a small percentage of travelers who start their trips during a congested time period are randomly selected for re-routing.

4. Travel Time (Msim) Stabilization – This is a Microsimulator based stabilization process in which travelers who experience a significant difference between their expected travel time and the travel times computed by the Microsimulator are selected and re-routed.

5. Equilibrium Convergence Process – This is a Microsimulator based stabilization process in which all the travelers are routed using the latest simulated link travel times. If the new travel time for a traveler is significantly different from the travel time on the original path, the travelers old plan is replaced with the new plan and the full plan set is re-simulated.

5.3.1 Router Stabilization

Router stabilization uses the link delay file generated by PlanSum using traditional volume to capacity equations to identify travelers who may benefit from re-routing. As mentioned earlier, two techniques were used to select the travelers for re-routing. The first technique was designed to identify and resolve potential bottleneck locations that are likely to cause gridlock within the Microsimulator. This is done by identify links and time periods where the volume to capacity ratio is significantly greater than 1.0. The PlanSelect program identifies these links and selects a random percent of travelers whose path travels through these links at the critical times of day for re-routing. This process was called V/C ratio stabilization.

V/C ratio stabilization is primarily focused on stabilizing the network performance. It does not directly address the user equilibrium objective. A second technique was used to address the user equilibrium condition. This procedure compared the trip duration (i.e., travel time) stored on the plan file with a re-estimation of this travel time based on the latest link delays. Since only selected plans are re-routed at any point in time, the link delays used to build a plan in the plan file may be significantly different from the current link delays. If the trip duration is significantly different (longer or shorter) than the current travel time for the path, the traveler is selected for re-routing. This process was called travel time stabilization.

The general Router stabilization process is shown in Figure 35. The PlanSelect program selects a random percentage of households based on the current selection criteria. The Router Process is applied and the link delays are updated based on a volume-delay equation. The process continues until the number of households qualified for re-routing becomes small or stops decreasing. For V/C ratio stabilization the selection criteria was based on the number of travelers that use links with a 15-minute V/C ratio greater than 1.5. For the travel time stabilization the selection criteria were based on travel time differences of plus or minus ten percent. In both cases only 50 percent of the qualified travelers were selected for re-routing at any given time. For example, when 15 percent of travelers used links with volume to capacity ratios greater than 1.5, only 50% of the those qualified travelers (i.e. 7.5 percent of all travelers) were selected for re-routing. The total number of re-routes was also limited to no more than 10 percent of the total travelers.

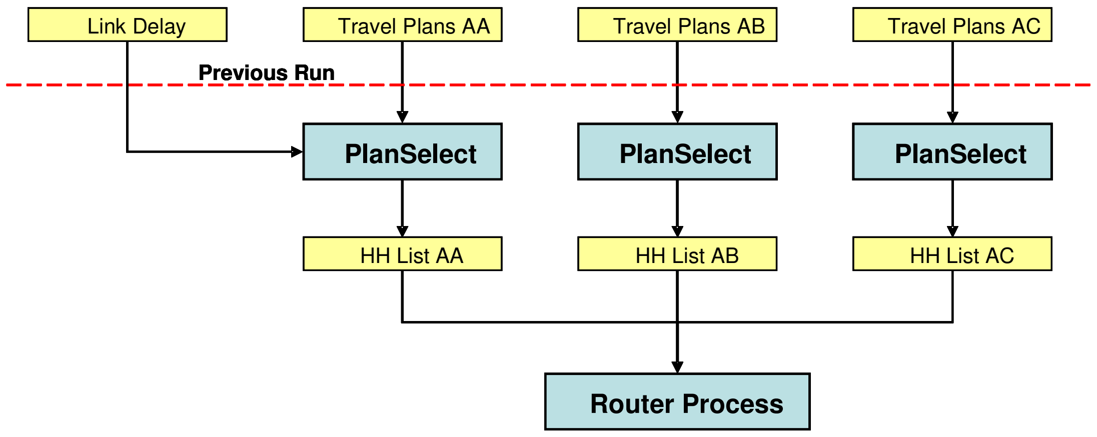


FIGURE 35: Router Stabilization

Two additional selection criteria were used for the travel time based selection method. These are the minimum time difference and the maximum time difference. These parameters constrain the selection to reasonable absolute time differences. The absolute difference must be greater than the minimum time difference for the trip to be selected. If the absolute difference is greater than the maximum time difference, the trip will always be selected. For the Portland study the household qualified if the travel time differences were two minute or more or greater than 30 minutes respectively.

At the end of the incremental loading process, about 11 percent of the plans had travel time differences of 10 percent or more from the travel times estimated using the final V/C-based link delays. The PlanSelect program randomly selected 50 percent of the qualified households for re-routing (i.e., 5.5 percent of total plans). The Router Process was run to re-route the selected households. The PlanPrep program merged the new plans with the original travel pans to create updated travel plans. The PlanSum was then used to generate the new V/C-based link delay file. The new link delay file and the new travel plans were used as the input to the PlanSelect program for the next stabilization run.

The Router-based travel time stabilization process described above helped in reducing the inconsistencies between the travel times computed from the link delay file and the plan duration by re-routing the selected travelers using the latest link delays. The stabilization process continued until the percentage of households that qualified for re-routing continued to decrease. The percentage of qualified travelers decreased from 11 percent to 4 percent after 10 stabilization runs.

5.3.2 Microsimulator Stabilization

Up to this point in the trip assignment process, the TRANSIMS Microsimulator has not been executed. The travel times used for path building were based on traditional volume-delay equations using simple volume-to-capacity ratios. The process generates a set of plans that are reasonably distributed to the network so that relatively few links are expected to be overloaded at any given time of day.

Volume-delay equations, however, do not accurately predict the travel times on arterial networks controlled by signals and stop signs. Vehicle interactions, lane changing, and turning movements also have significant impacts on roadway performance. Figures 38 and 39 compare speed-flow relationships generated for the same set of trips using V/C ratios and the Microsimulator. The speed-flow curve from the Microsimulator is reasonably close to the V/C speed-flow curve for the freeway facility. The speed-flow curve for the arterial is significantly different from the V/C curve. On arterial facilities, the simulated speeds are relatively stable as the volume approaches the saturation flow rate and then drops steeply when an intersection breaks down. The primary reason for this discrepancy is that the Microsimulator takes into account the delays that occur at intersections due to traffic controls, network congestion and vehicle interactions while the V/C computation ignores the intersection delays.

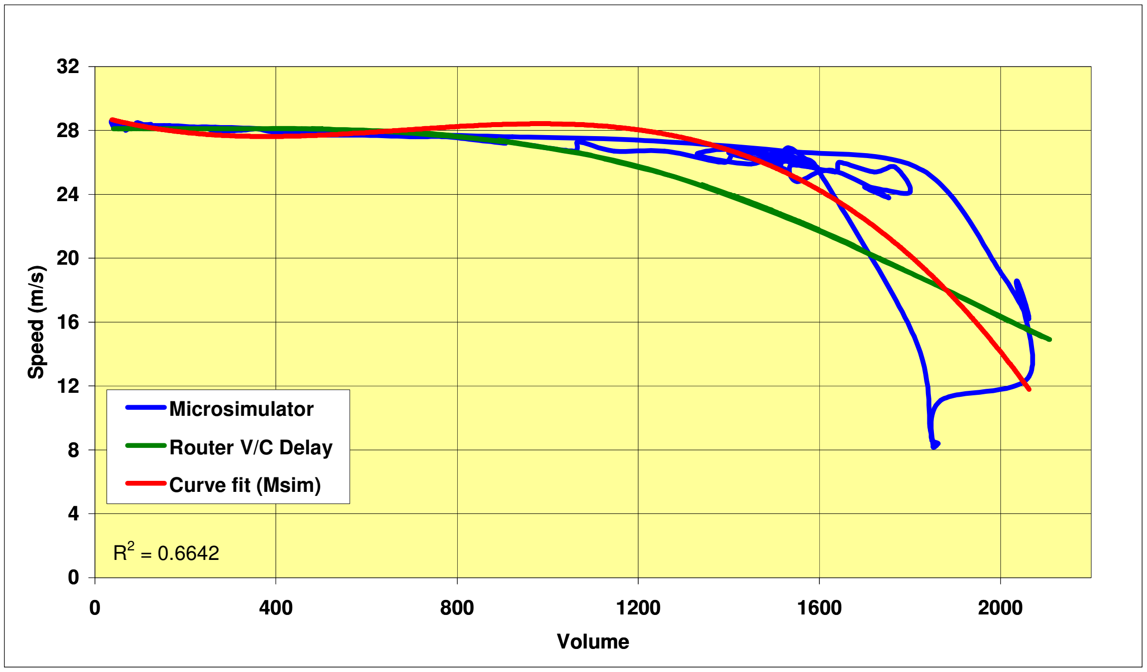


FIGURE 36: Speed-Flow Relationship for Freeway Facilities

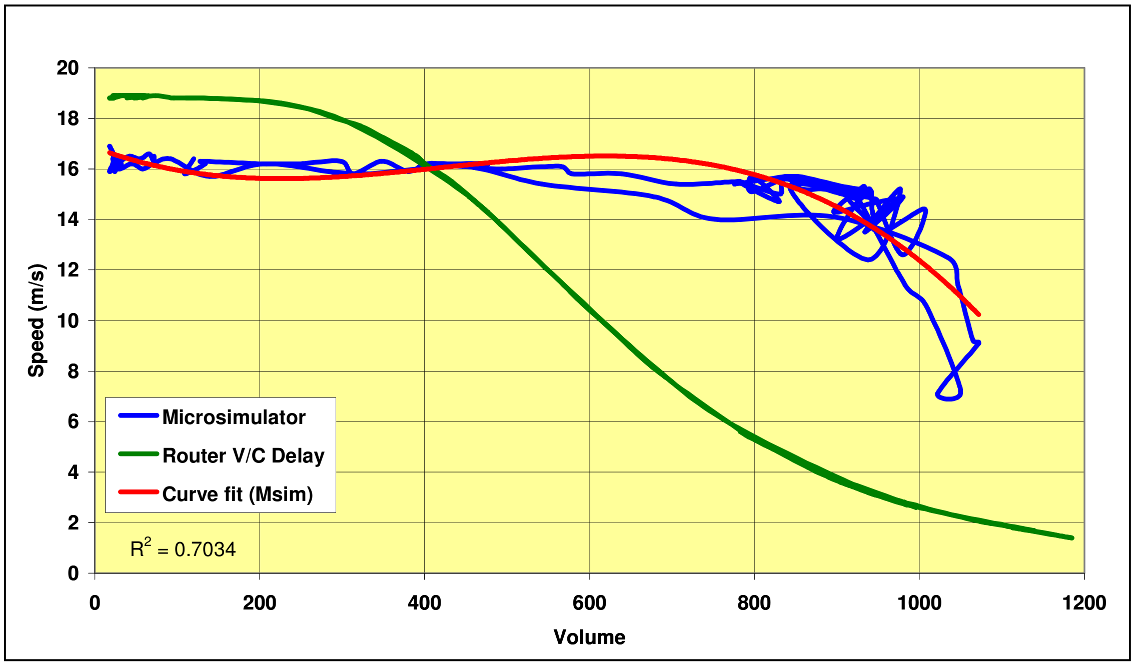


FIGURE 37: Speed-Flow Relationship for Arterial Facilities

The flow relationships clearly indicate that the V/C link delays cannot match the intersection link delays generated by the Microsimulator. The Microsimulator needs to be applied to correct the faulty assumptions about capacity and travel time used to generate the plans. The Microsimulator stabilization loads trips onto the network and corrects the travel time assumptions. This needs to be done in an orderly and regimented way in order to avoid illogical side affects and maintain processing efficiency. The critical situation to avoid is a bottleneck that creates a cascading queue that results in system gridlock. Once the system reaches total gridlock, the 15-minute travel times reported by the Microsimulator have very little meaning. The Microsimulator also takes significantly longer to process gridlock conditions.

The Microsimulator stabilization process involves all three of the batch processing modules described in Chapter 4 (Router Process, Time Sort Process and the Msim Process). The process basically begins by simulating the plans generated by the Router stabilization until the network becomes severely congested. This typically occurs during the AM Peak period. Snapshot files generated by the Microsimulator are reviewed using the Output Visualizier to check whether the Microsimulator produces gridlock and cascading queues. When the simulation results look unreasonable, the Microsimulator run is terminated and the stabilization process begins. Gridlock conditions can also be detected using the Microsimulator step processing time. When the time required to process one simulation second increases by a factor of ten, the Microsimulator can be stopped.

The stabilization process randomly selects plans to re-route based on the time-of-day when the Microsimulator is most congested. The households are re-routed using the travel times from the Microsimulator, merged with the rest of the plans, and re-simulated. This process continues until the queues subside and all of the trips for the day are simulated. Figure 38 shows the general Microsimulator stabilization process.

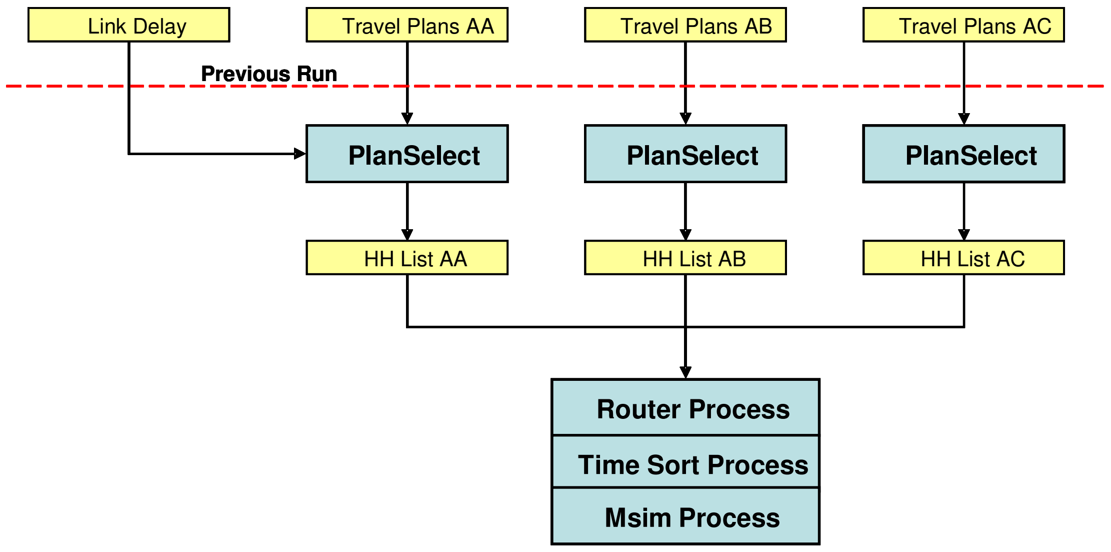


FIGURE 38: Microsimulator Stabilization

For the Portland application, two microsimulation iterations were needed produce reasonable levels of congestion during the AM peak period. This was followed by two iterations to address the mid-day time period and three iterations to clear major traffic problems during the evening peak period. A total of ten time based stabilization runs were performed to address illogical levels of traffic congestion. During this process, the simulation results were visualized after each run and the time periods that warranted re-routing were identified. The PlanSelect program was used to select travelers within the identified time periods. The results looked reasonable in the Output Visualizer, but at the end of the tenth stabilization run, about 30 percent of the households were qualified for re-routing based on travel time differences. The fact that 30 percent of the households were qualified indicated that there were still a significant number of travelers who experienced 10 percent or more difference between the travel time coded in their plan and the travel time computed from the link delay. The stabilization process continued until the percentage of qualified households no longer decreased. Fifteen travel time stabilization runs were performed using the Router-Microsimulator combination to achieve a stable result. The percentage of households that qualified for re-routing after 15 more iterations was six percent. At this point, a total of 25 simulations were performed.

5.3.3 Equilibrium Convergence

The final step in the trip assignment process is equilibrium convergence. This step differs from the travel time stabilization in one crucial way. In the travel time stabilization a traveler is only selected for re-routing if the travel time on their existing path changes significantly. There is no direct consideration of other paths in the selection process. What the equilibrium convergence process does is re-route all of the travelers using the latest link delays from the Microsimulator. The travel times on these new paths are compared to the travel times in the original paths. If the times are significantly different, the new plan qualifies to replace the old plan in the full plan set. For the Portland study the travel time change between successive runs was considered to be significant if the percent change in travel time was greater than 10 percent.

A slight modification to the routing routine is required to implement this process. As shown in Figure 39, the complete list of households assigned to each partition is re-routed using the link delays from the previous iteration. The output of the Router is the minimum travel time path for each traveler. The PlanCompare program compares the total travel time on the minimum travel time path to the total travel time on the original path. If the travel time for a given traveler is significantly different (greater than 10 percent), user equilibrium has not been achieved and plan adjustments should be considered.

The same five selection criteria described in the Router stabilization section are applied here. The percent time difference, the minimum time difference, and the maximum time difference are used to determine if the time difference is significant enough for consideration. The selection percentage and the maximum percent selected are then used to determine which of the qualified plans will replace the existing plan in the full plan set. The updated plan set is re-simulated and new link delays are generated. The process continues until the number of plans with significant travel time differences is small.

For the Portland study, a total of 20 equilibrium convergence stabilization runs were made by re-routing all travelers and selecting the replacement plans. Experience with Portland has shown that TRANSIMS cannot reach absolute user-equilibrium with this method. In other words, there will always be some travelers who can improve their travel time by a small amount. From a theoretical perspective, user-equilibrium cannot be achieved on a network with intersection controls and integer-based (i.e., whole vehicle) trips. The best that has been achieved to date using this method still leaves about two percent of the trips unresolved.

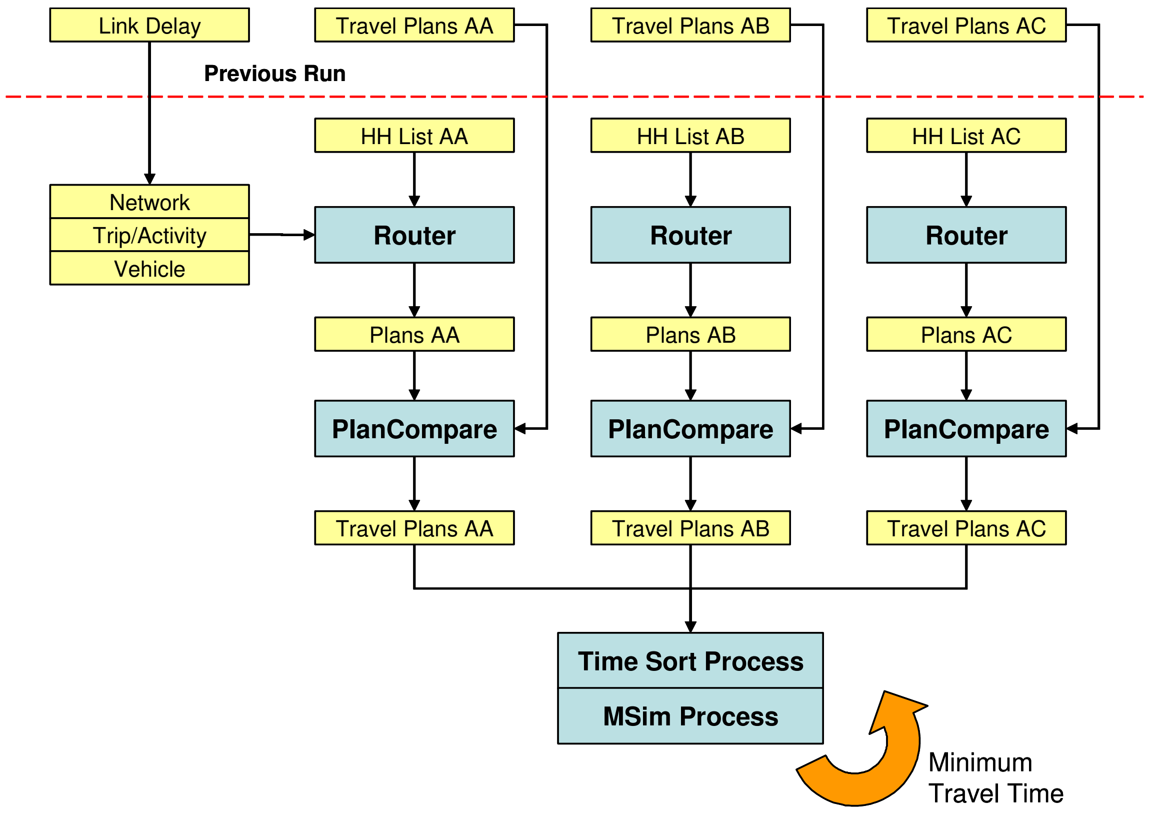


FIGURE 39: Equilibrium Convergence

5.3.4 Network Calibration

In addition to re-routing trips to avoid congestion bottlenecks, the Microsimulator stabilization process is the time when network coding issues are addressed. There are a number of data items that have significant impact on the performance of the Microsimulator, but no impact on the Router. These include pocket lanes, lane connectivity, lane changing, cross street traffic at unsignalized intersections, and signal timing and phasing plans at signalized intersections. If this is the first time a particular network has been simulated, it is likely that many or all of these data items will need to be reviewed and refined to ensure the network behaves appropriately. In the Portland study, it took about two to three weeks to review and edit the TRANSIMS network based on the Microsimulator results.

The Output Visualizer and the Network Editor are useful in this process. The snapshot file generated by the Microsimulator is reviewed in the Output Visualizer. The Visualizer will show the queues growing by time of day. The user can zoom in on the queue and attempt to identify the intersection where a cascading queue begins. Once the source of the problem is identified, the Network Editor can be used to review the network and make the necessary changes. Aerial pictures of Portland were used extensively to validate and modify the network. Figure 40 shows how the Output Visualizer can be used to zoom in to a particular intersection to visualize queues. Figures 41 shows an example of cascading queues and system gridlock.

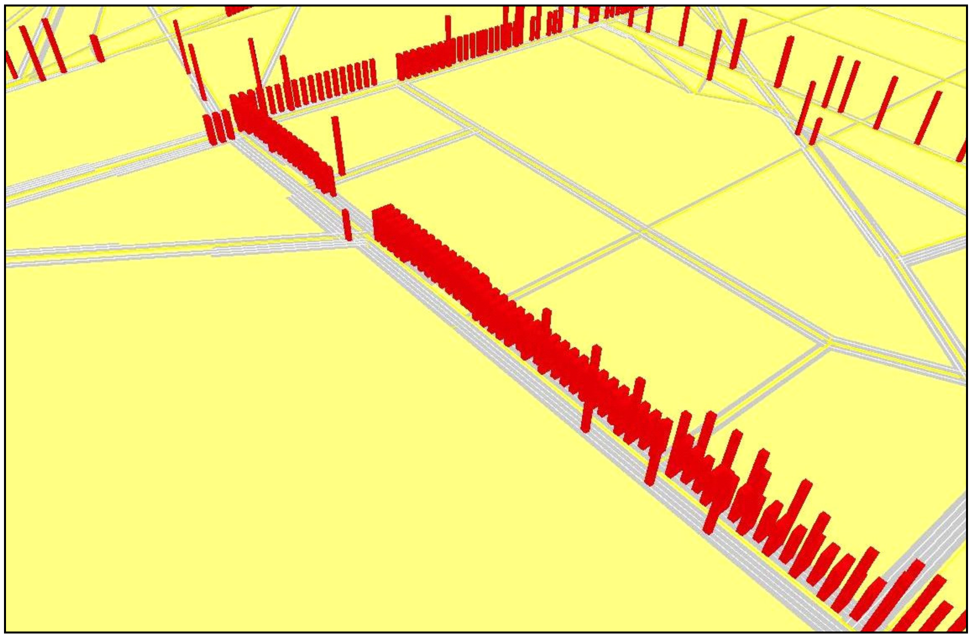


FIGURE 40: Visualizing Network Coding Errors

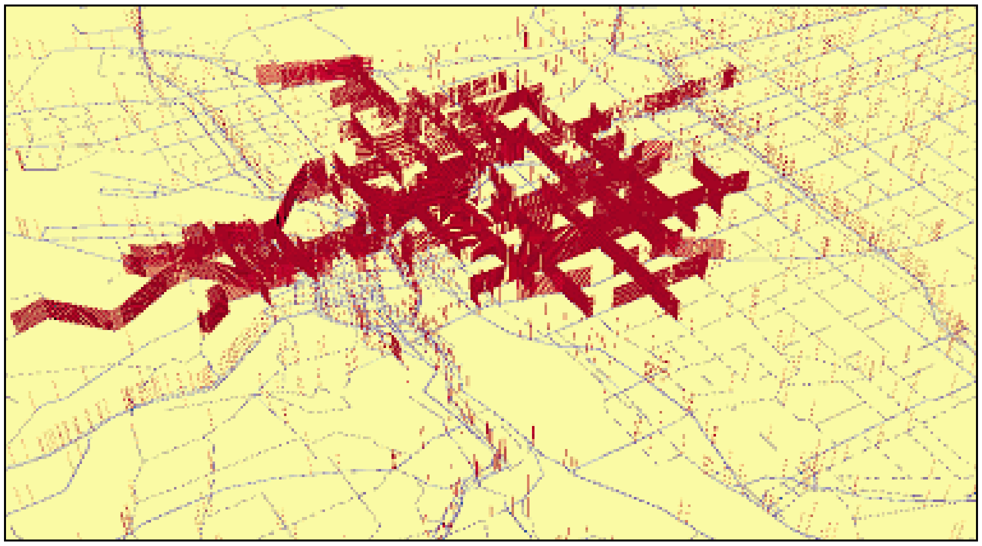


FIGURE 41: Cascading Queues and System Gridlock

When major edits are made to the network (such as removing a turning movement, deleting a link, making a two-way street to one-way link, etc.) all travelers who are affected by the network change need to be re-routed to adjust their travel plans. The PlanSelect program can be used to identify the travelers who use particular links or nodes. The identified households can then be re-routed around the network change.

5.3.5 Lost Vehicles

Another aspect of the simulation that needs to be reviewed, but may not be obvious from a snapshot file, is “off plan” or lost vehicles. These are vehicles that fail to make a necessary maneuver due to congestion or lane connectivity issues. If a vehicle goes “off plan”, the Microsimulator removes it from the simulation, prints an error message in the log file, and adds a record to the event file (if the user requests one). The MsimSum and EventSum programs can be used to identify locations where large numbers of vehicles get lost. The coding at these locations should be reviewed in the Network Editor. The vehicles also get lost of go “off-plan” if the vehicle gets stuck at one place for more than the maximum permissible waiting time that is set defined. The vehicles typically get stuck at the same location when cascading queues result in severe grid locaks as shown in figure 42 above.

The log file generated by the Microsimulator contains a number of error messages that are important in understanding and resolving network coding issues. This file is typically large and cannot be easily reviewed or manipulated with standard software tools. The MsimSum program was developed to read through the log file to find and summarize error messages of special interest. These include blocked intersections, turning maneuver problems, and travel schedule and parking issues. In addition to printed reports, the program generates data files that can be merged with network link and node information to graphically display locations with a significant number of problems. Figure 42 shows how these data files can be used to visualize the network locations where significant numbers of vehicles are lost due to network coding issues.

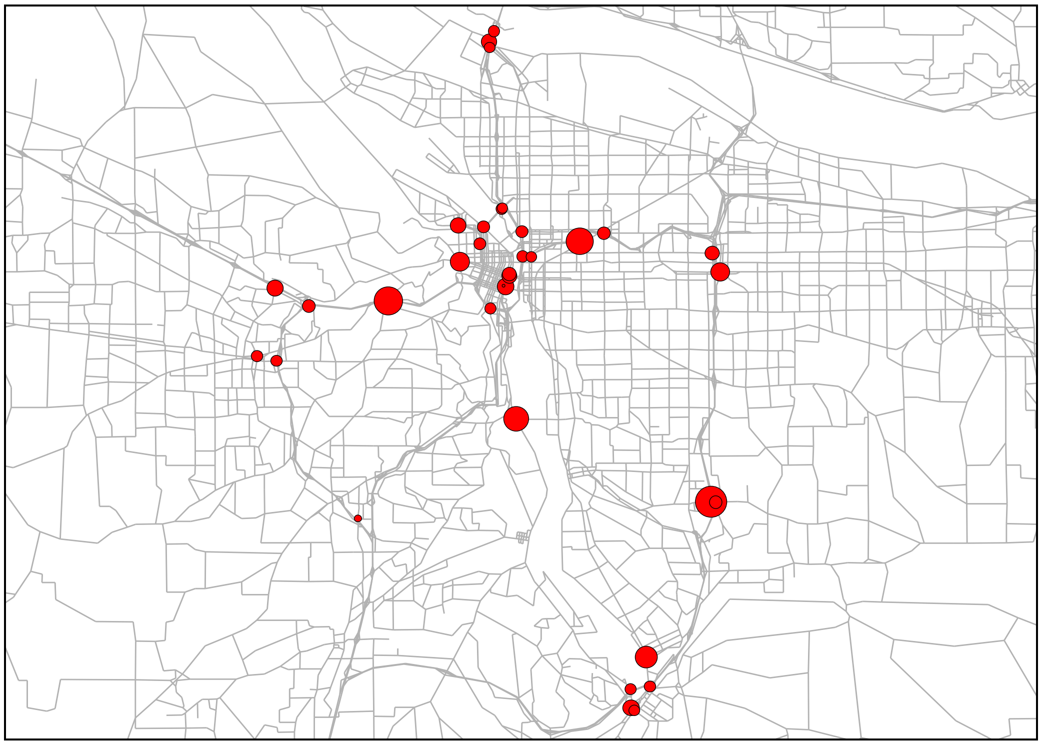


FIGURE 42: Lost Vehicle Locations

5.4 Performance Measures

The TRANSIMS software produces large quantities of data that could be used for evaluating the system performance in dimensions beyond the scope and capabilities of traditional travel demand forecasting tools. For example, the simulation data can be used for studying the travel time and speed variations by time of day. The data files generated by the Microsimulator are generally huge (0.5 to 1.2 gigabytes) and cannot be easily handled by commonly used text editors and other software tools. In order to make the data readily usable and meaningful the PlanSum and LinkSum programs were developed to generate reports and data that can be used to measure the performance of the runs and compare one run to another. Run comparisons are particularly helpful in quantifying the stopping criteria. They also enable compares of multiple applications to a common base.

5.4.1 V/C Ratios by Time of Day and Facility Type

The volume to capacity ratio plots categorized by time of day and facility type are a good indicator of how the trips are loaded onto the network. Sample V/C ratio plots for different percentiles of freeway and arterial lane-miles are shown in Figures 45 and 46. The plots indicate that more than 90 percent of the freeway lanes-miles have V/C ratios less than 1.0 for most of the time periods. Only 5 percent of the AM Peak Period lane miles and 10 percent of the PM Peak Period lanes miles experience over capacity conditions. An even smaller percentage of the arterial lane-miles experience over capacity conditions during the morning and evening peak periods.

It is important to note that these charts have significantly different meaning if they are calculated based on Router demands versus Microsimulator volumes. The charts shown in Figures 43 and 44 were calculated from the PlanSum link delay file. These volumes represent the demand for a given roadway at a given time of day. That means that these V/C ratios are basically the same as V/C ratios generated by traditional models. If the calculations were made based on the link delay file generated by the Microsimulator, the volumes are constrained by the simulation throughput. Since the Microsimulator does not consider the user-provided capacity values in calculating link performance, capacity is unlikely to accurately represent Microsimulator throughput. A chart generated from Microsimulator volumes would show the relative accuracy of the user-provided capacity values. If it were compared to the demand chart, it would show how much of the demand the network was able to accommodate at any given time of day.

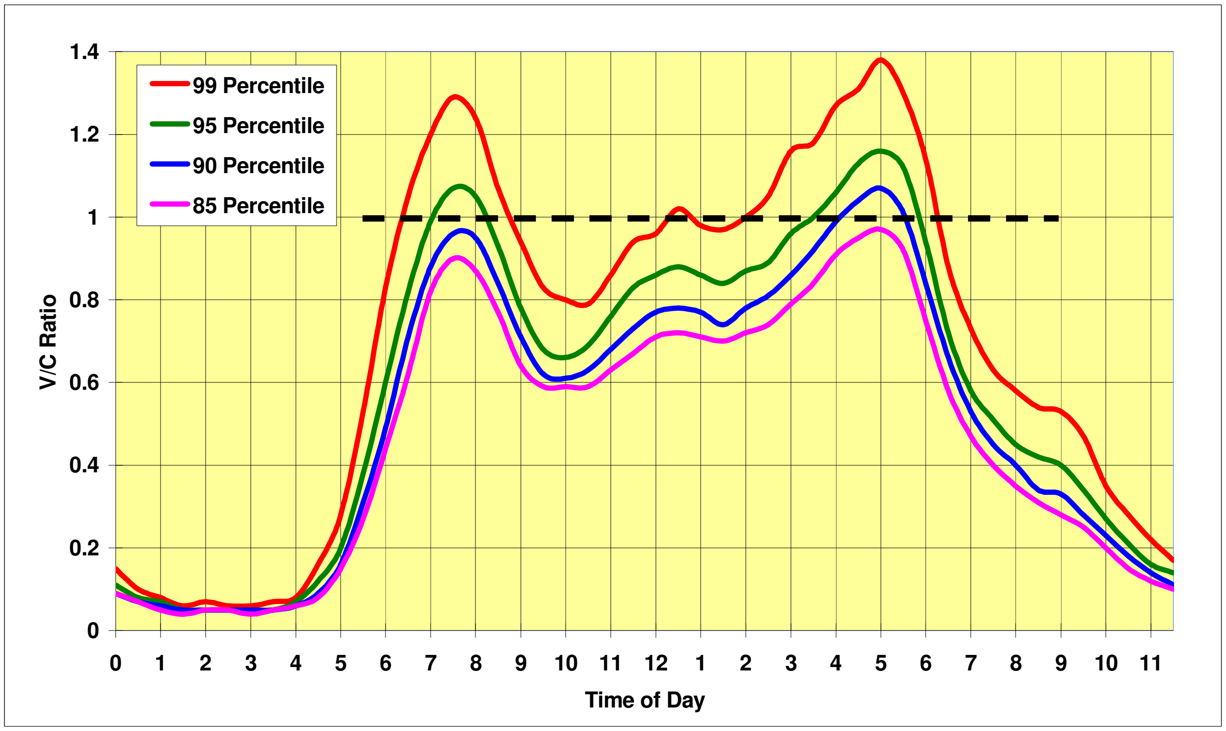


FIGURE 43: Volume-to-Capacity Ratios on Freeway Lane-Miles

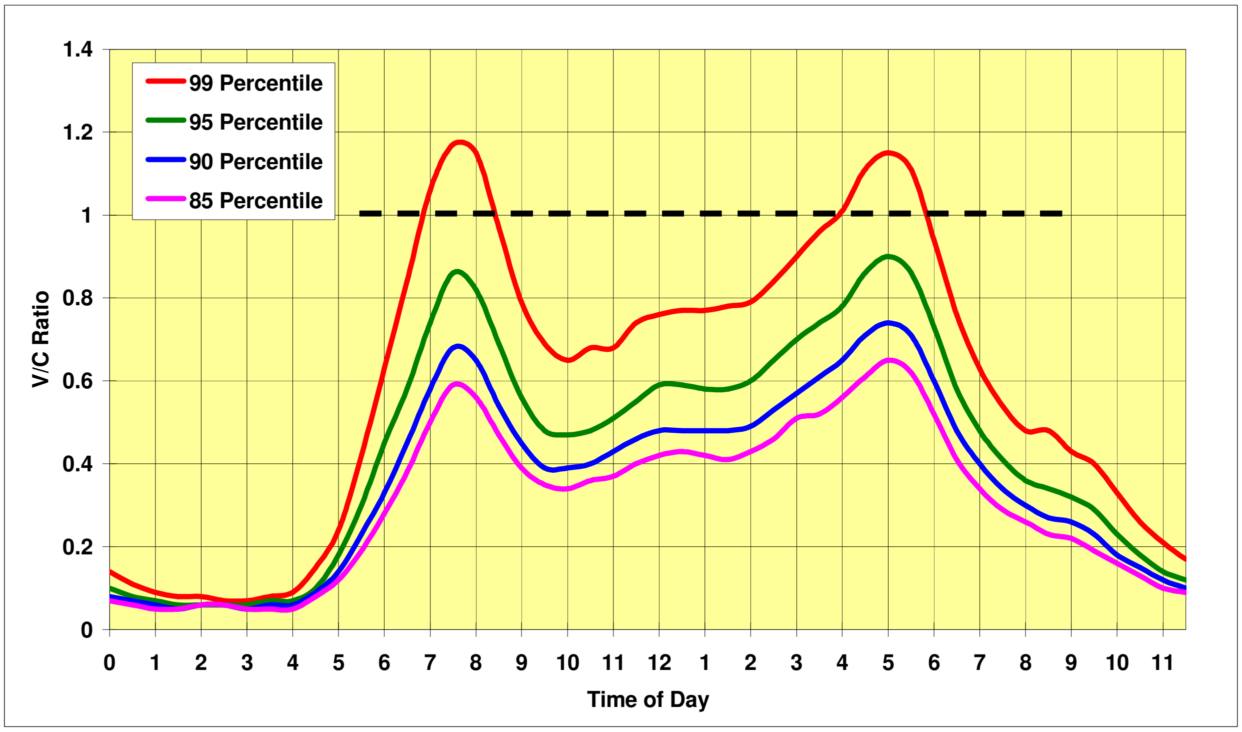


FIGURE 44: Volume-to-Capacity Ratios on Arterial Lane-Miles

5.4.2 Travel Time Ratio

The travel time distribution by time period is a useful measure for gauging network congestion levels. The LinkSum program defines travel time ratios as the loaded travel time divided by the base travel time. The user can specify a minimum travel time to control the impacts of short links on the travel time ratios. Links with travel times less than the minimum travel time are not included in the distribution. A sample travel time ratio distribution by time period is shown in the Table 3. Travel time ratios of 2.0 or above indicate that the run is congested. Note that this measure avoids the independence problem of V/C ratios within TRANSIMS. It also can be generated using V/C-based speeds or simulated speeds.

In both cases the base travel time is calculated from a weighted combination of free flow speed and speed limit. In TRANSIMS free flow speeds are used in the Router to estimate the travel time on links with no other traffic but with traffic control delays. TRANSIMS uses speed limit in the Microsimulator to define the maximum speed a vehicle can traveler on the link. The Microsimulator then reduces this travel time based on the specific traffic controls at the end of the link and vehicle interactions. A weighted combination of free flow speed and speed limit provides a reasonable estimate of the maximum speed that the Microsimulator will generate on a given link.

TABLE 3: Travel Time Ratio - Percentile Distribution by Time Period

```
                    Loaded / Base Travel Time Ratio
           -------------Percentile Distribution by Time Period----------      Lane
Time Period   50%    65%    70%    75%    80%    85%    90%    95%    99%       KM

 630- 700    0.93   0.96   0.98   1.01   1.07   1.15   1.28   1.55   2.28     8589
 700- 730    0.92   0.96   0.97   1.00   1.05   1.13   1.26   1.53   2.31     9714
 730- 800    0.92   0.96   0.98   1.01   1.06   1.14   1.27   1.54   2.42    10234
 800- 830    0.92   0.96   0.98   1.01   1.06   1.14   1.28   1.56   2.44    10193
 830- 900    0.92   0.96   0.98   1.01   1.05   1.13   1.26   1.52   2.29     9898
1430-1500    0.92   0.96   0.97   1.00   1.05   1.12   1.25   1.51   2.25    10148
1500-1530    0.91   0.96   0.97   1.00   1.04   1.12   1.25   1.52   2.28    10302
1530-1600    0.91   0.95   0.97   1.00   1.04   1.12   1.25   1.52   2.26    10460
1600-1630    0.91   0.95   0.97   1.00   1.04   1.12   1.25   1.51   2.31    10534

Total        0.92   0.96   0.98   1.01   1.05   1.14   1.27   1.54   2.29   355337
```

5.4.3 Volume and Travel Time Changes

The LinkSum program can also be used to compare two link delay files to quantify the changes in volume or travel time by time of day. These measures are useful for quantifying network stability or comparing alternatives. They may be generated from link delay files produced by two Microsimulator runs, two Router applications, or a comparison of the Microsimulator volumes to the Router demands. In all cases the change reports produce a distribution of differences by time of day and the magnitude of the change. Sample volume and travel time change reports are shown in Tables 5 and 6. These reports were used in the Portland study as the network stability stopping criteria. The overall objective was to approximate user-equilibrium while at the same time achieving network stability. In this case, network stability was defined as minimal changes in travel time and volume by time of day between two successive Microsimulator runs.

TABLE 4: Travel Time Changes

	Time Change * 100 / Previous Time									
	Percentile Distribution by Time Period									Lane
Time Period	50%	65%	70%	75%	80%	85%	90%	95%	99%	KM
630- 700	2.2	3.8	4.6	5.8	7.2	9.2	12.4	18.4	33.5	8684
700- 730	2.0	3.5	4.3	5.2	6.4	8.3	11.3	16.7	30.9	9785
730- 800	2.0	3.5	4.2	5.1	6.2	8.2	11.4	17.4	32.0	10291
800- 830	1.9	3.4	4.0	5.0	6.2	8.0	11.0	17.3	33.1	10255
830- 900	2.0	3.4	4.1	5.0	6.1	7.9	11.0	16.4	31.8	9966
900- 930	1.9	3.2	4.0	4.9	6.2	8.1	11.1	17.3	31.6	9588
1400-1430	2.0	3.5	4.3	5.2	6.3	8.0	10.8	16.7	31.1	9992
1430-1500	2.0	3.3	4.0	5.0	6.1	7.9	10.6	16.3	31.3	10232
1500-1530	2.0	3.3	4.0	5.0	6.3	8.2	11.1	16.5	30.0	10363
1530-1600	2.0	3.3	4.0	4.9	6.2	7.9	11.0	16.6	30.7	10528
1600-1630	2.1	3.4	4.1	5.0	6.1	7.8	10.7	16.5	30.0	10580
Total	2.1	3.6	4.3	5.3	6.6	8.5	11.5	17.5	33.1	358054

TABLE 5: Volume Changes

	Volume Change * 100 / Previous Volume									
	Percentile Distribution by Time Period									Lane
Time Period	50%	65%	70%	75%	80%	85%	90%	95%	99%	KM
630- 700	2.9	4.5	5.4	6.6	7.7	9.4	11.8	16.1	24.2	8684
700- 730	2.8	4.5	5.4	6.4	7.6	9.0	11.1	14.8	21.9	9786
730- 800	2.9	4.6	5.5	6.6	7.7	9.5	11.6	15.7	23.5	10291
800- 830	3.0	4.8	5.7	6.6	7.7	9.3	11.5	15.3	23.5	10255
830- 900	2.9	4.8	5.6	6.6	7.8	9.6	11.9	15.9	24.1	9966
900- 930	2.8	4.3	5.2	6.1	7.3	8.9	11.3	15.2	23.4	9588
1400-1430	2.7	4.3	5.0	5.9	6.9	8.3	10.5	14.2	22.2	9992
1430-1500	2.7	4.3	5.1	6.0	7.2	8.6	10.9	14.2	22.7	10232
1500-1530	2.8	4.5	5.3	6.1	7.3	8.7	10.8	14.2	22.5	10363
1530-1600	2.8	4.3	5.1	6.0	7.1	8.4	10.4	13.7	21.2	10528
1600-1630	3.3	5.3	6.2	7.3	8.6	10.5	13.2	17.8	27.4	10581
Total	2.9	4.6	5.5	6.5	7.6	9.3	11.7	15.9	25.0	358060

Chapter 6 Validation of Results

Chapter 5 described the traffic assignment process that was adopted for the Portland TRANSIMS application. The TRANSIMS Router and Microsimulator were used to build paths and simulate traffic flow through the network. An iterative process was developed to efficiently stabilize the traffic volumes and speeds while at the same time minimizing the travel times for each trip. Review of the simulation results using the Output Visualizer showed that the Microsimulator generated logical levels of congestion by time of day.

These conclusions were confirmed using traditional and non-traditional validation measures. Traditional measures included comparisons of assigned volumes to observed traffic counts on screenlines, facility types, area types and volume group. Some non-traditional measures included distributions of traffic and speeds by time of day. It was also necessary to calibrate and validate the simulation model to accurately replicate the traffic throughput observed in Portland. This chapter documents the results of these efforts.

6.1 Microsimulator Calibration and Validation

The initial validation results using daily screenline counts showed that the volumes generated by the Microsimulator were less than the observed counts by 10 percent or more. One of the major reasons for the under assignment was the number of lost vehicles generated by the Microsimulator. Chapter 5 described the steps taken to address lost vehicle issues through network coding corrections and iterative re-routing. These measures addressed many of the localized problems, but they did not have much impact on the overall screenline comparisons. The Microsimulator was still loosing a significant number of trips even though the volumes were underestimated.

In addition, small changes or adjustments to the network or the signal timing had significant impacts on the simulation results. Minor edits to the signal timing plans at a few intersections could change a reasonable simulation into severe gridlock from cascading queues. Since the Portland network in 1994 was not overly congested, the Microsimulator was much more sensitive to small volume increases on critical links than was expected from local experience. The prospect of simulating future traffic levels on a future year network with relatively few improvements was daunting.

The simulation instability and the lost vehicle problem suggested that adjustments to the Microsimulator parameters were needed to increase the overall throughput of arterials and freeways. A series of detailed sensitivity tests were performed to understand the behavior of the Microsimulator and determine appropriate adjustments. The results of these tests are described below.

6.1.1 Lost Vehicle Problems

In the early Microsimulator runs, as much as 10 percent of the trips could not be completed. The problems were generally associated with short multi-lane roadway links where vehicles needed to make several lane changes to be in position for a turning movement at the next intersection. As the traffic volumes increased, the vehicle was unable to change lanes because there were too many other vehicles in that lane already. As a result, the vehicle reached the intersection in a lane that did not have the necessary lane connectivity to continue the trip so the vehicle went “offplan” and was lost. Figure 45 shows how the number of lost vehicles increased significantly during peak periods. In other words, lost vehicles are a direct result of congestion.

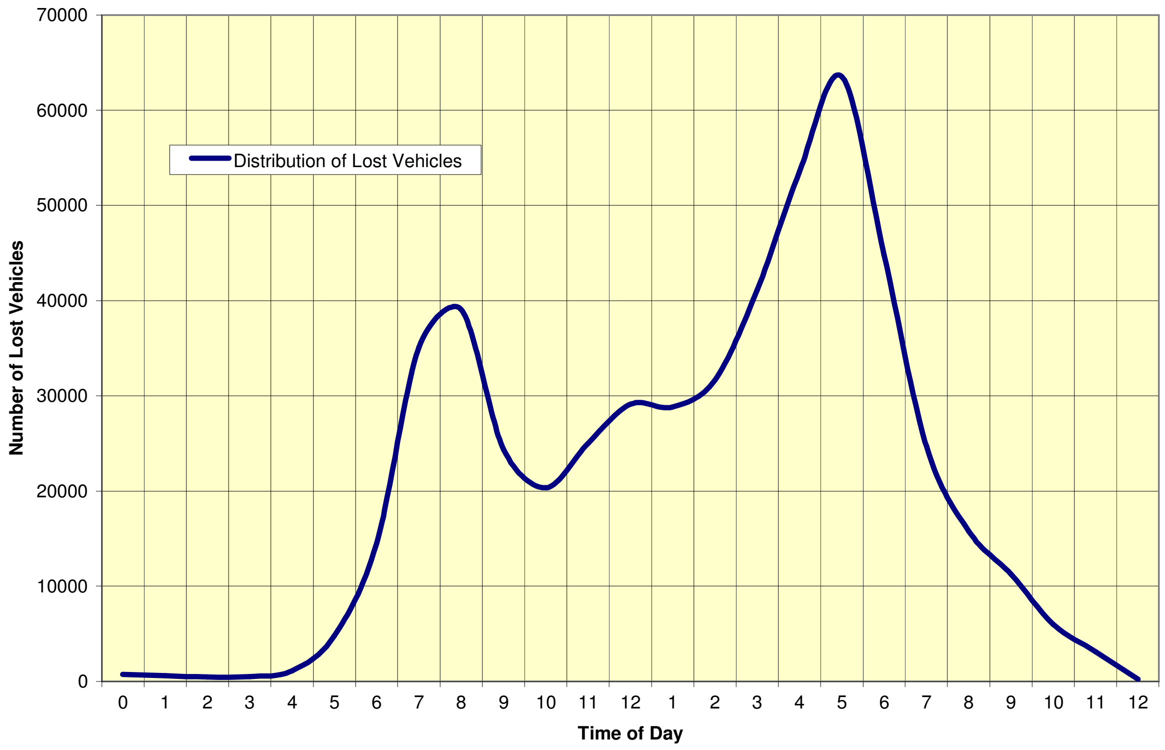


FIGURE 45: Distribution of Lost Vehicles

Congestion, however, did not explain all of the lost vehicle issues. A major portion of the trips were lost within the first few seconds of being loaded onto the network. Figure 46 shows that about 50 percent of the lost vehicles were lost within the first 3 minutes of their trip and about 15 percent of the trips were lost within the first few seconds after the vehicle was loaded onto the network. An investigation found that the problems increased when the parking lot was close to an intersection. When a vehicle leaves the parking lot, TRANSIMS places the vehicle in the left-most lane if space is available. If the vehicle is then required to make several lane changes to get into the proper lane to continue its trip, there may not be sufficient distance or time to complete these maneuvers. This can cause vehicles to be lost even when there is no other traffic on the roadway.

A number of potential solutions to the lost vehicle problem were evaluated. These included:

1. Arbitrarily increasing the link length on short links,
2. Adding additional lane connectivity at intersections to eliminate the need for lane changing,
3. Adjusting parking lot locations, and
4. Setting the "never lost" parameter for all vehicle types (not just buses).

These changes, of course, made the network less "realistic" and therefore added a different type of error to the analysis process. The trade-off between real-world coding and simulation approximation needed to be made carefully. A set of systematic coding rules were considered in order to apply the adjustments uniformly and consistently for both existing and future networks. In the end, these solutions were rejected as too cumbersome and problematic. A better solution was needed.

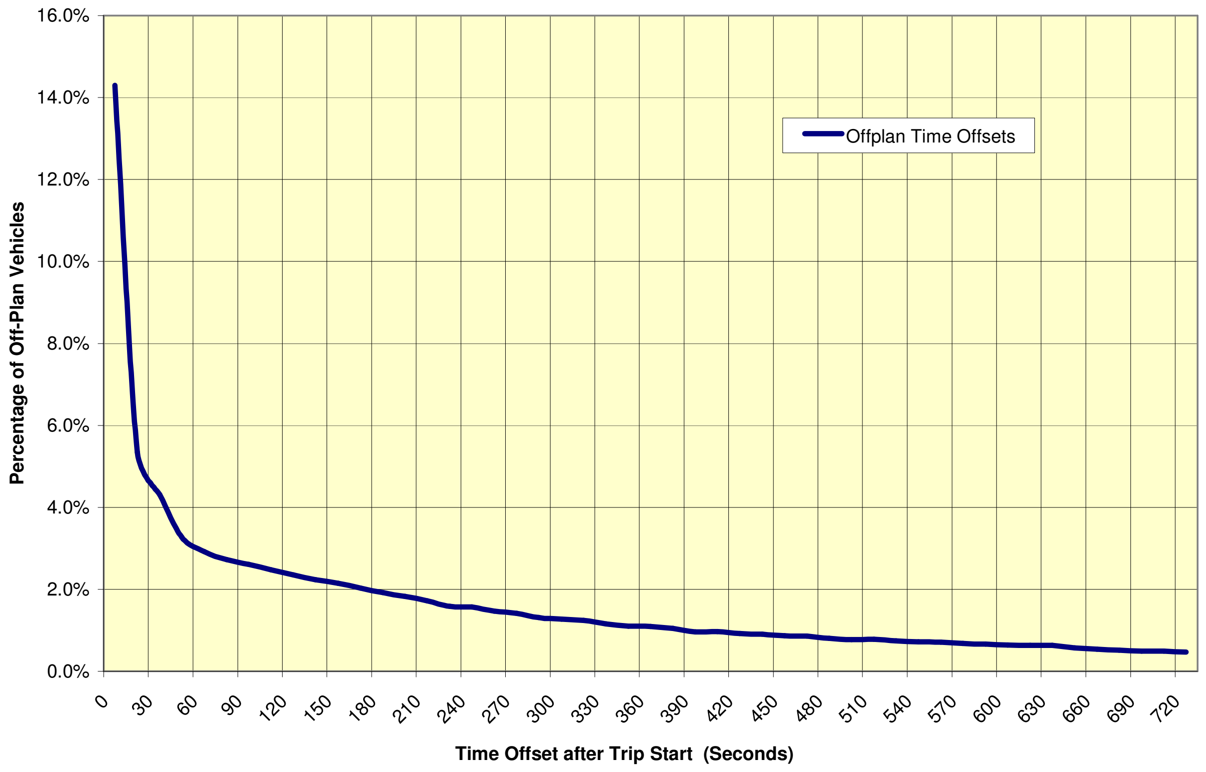


FIGURE 46: Time after the Trip Start when the Vehicle is Lost

6.1.2 Network and Activity Rescaling

Analysis of the simulation results showed that the default lane changing algorithm within the TRANSIMS Microsimulator was not aggressive enough to allow all the necessary lane changes required by the vehicles to complete their turning movements. The fact that the Microsimulator considered a lane change in a given direction once every other second made the distance and time required to change multiple lanes difficult to achieve. More frequent lane changing opportunities were needed.

The Microsimulator does provide a number of configuration keys that permit the user to control the simulation fidelity. Adjustments to most of these parameters had little affect. It was discovered that a number of hardwired relationships within the source code made changes to the adjustment parameters problematic. For all practical purposes, the Microsimulator cell size was fixed at 7.5 meters, the time step was fixed at one second, the speed increments were fixed by cells per second, and the gap acceptance and lane changing rules were based on a fixed number of cells. The only parameter that could be changed safely was the vehicle size (number of cells).

Given these constraints, the only practical solution that was available to increase the fidelity of the Microsimulator was to increase the scale of the input datasets. By artificially increasing the link lengths and the time of day information, the Microsimulator could use the fixed relationships to simulate the real world at a higher level of fidelity. A number of scaling factor and vehicle size combinations were tested to determine the level of increase that resulted in realistic and stable conditions. Scaling factors of 1.25, 1.5, 2.0, and 3.0 were tested along with one cell vehicles and two cell vehicles. In the end, a scaling factor of 2.0 with two cell vehicles produced the best results.

Figure 47 shows how the network scaling impacted the cell size, time steps, and lane changing behavior from the Microsimulator's perspective. Since the links were twice as long, they now contained twice as many cells. Since the activity times, signal timing plans, and other time-based attributes were also doubled, each one second time step in the Microsimulator represented 0.5 seconds in the real world. This made the speeds (cells per time step) equal in size and impact to the original network. Since the vehicles were made two cells long, they required the same amount of physical space on links and in queues. The primary benefit was that the Microsimulator could now consider lane changing twice as often and the gap acceptance rules were less generous. In other words, the travelers were more aggressive by accepting smaller gaps in traffic.

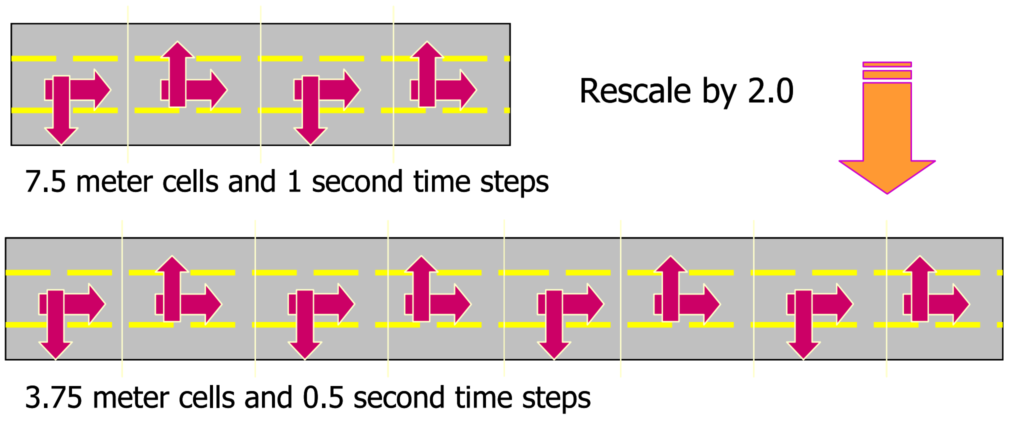

Rescale by 2.0
7.5 meter cells and 1 second time steps
3.75 meter cells and 0.5 second time steps

FIGURE 47: Microsimulator Rescaling

One of the negative side effects of rescaling was that the Microsimulator needed to simulate traffic for 48 hours to generate 24-hour statistics. The total computer processing time, however, was approximately the same. By creating a less congested, more stable simulation, the rescaled approach did not have the cascading gridlock conditions that significantly slowed down the Microsimulator. The overall process also required fewer feedback iterations to stabilize the results.

After the network and the activities were rescaled, the Microsimulator runs were more stable and the validation results improved dramatically. The total number of lost vehicles in the validated model run was less than 70,000 vehicles (about 1.3 percent of total trips). This was a major improvement over the 520,000+ vehicles (about 10.4 percent of total trips) that were getting lost during the early stages of model development. It also significantly improved the screenline statistics. These results are presented in the next section.

6.2 Traffic Count Validation

The assignment results were validated against observed traffic counts. The daily and PM peak period counts used by METRO to calibrate their regional model were used to validate the TRANSIMS process. Counts on 410 directional links were used for validation purposes. These links were grouped into the 64 screenlines used by METRO. In addition to screenlines, the Validate program generates summary reports by functional class, area type, district, and volume level. Tables 6 and 7 show the functional class and volume level summaries based on daily volumes. Tables 8 and 9 show the PM peak period results.

TABLE 6: Daily Validation by Volume Level

Summary Statistics by Volume Level

Volume Level	Num. Links	Total Volume	Total Count	Difference	% Diff	Avg. Error	Avg % Error	Std. Dev.	% RMSE
0 to 1000	13	7152	7696	-544	-7.1	612	103.5	422	124.1
1000 to 2500	54	75172	89590	-14418	-16.1	1421	85.7	845	99.4
2500 to 5000	85	206323	342307	-135984	-39.7	2438	60.5	1464	70.5
5000 to 7500	58	226082	361719	-135637	-37.5	3737	59.9	2161	69.1
7500 to 10000	35	219916	304438	-84522	-27.8	4077	46.9	2001	52.1
10000 to 25000	110	1701242	1555812	145430	9.3	4073	28.8	3129	36.3
25000 to 50000	19	802918	708268	94650	13.4	5970	16.0	4054	19.2
50000 to 75000	34	2244400	2150614	93786	4.4	7396	11.7	5692	14.7
75000 to 100000	2	185029	163400	21629	13.2	10814	13.2	975	13.3
TOTAL	410	5668234	5683844	-15610	-0.3	3624	26.1	3234	35.0

TABLE 7: Daily Validation by Functional Class

Summary Statistics by Functional Class

Functional Class	Num. Links	Total Volume	Total Count	Difference	% Diff	Avg. Error	Avg % Error	Std. Dev.	% RMSE
Freeway	52	3025656	2874830	150826	5.2	6427	11.6	5313	15.0
Principal Arterial	24	505135	409706	95429	23.3	5091	29.8	3978	37.5
Major Arterial	104	1312825	1143236	169589	14.8	3525	32.1	2739	40.5
Minor Arterial	82	401095	507048	-105953	-20.9	2900	46.9	2268	59.4
Collector	130	139455	462788	-323333	-69.9	2736	76.9	2064	96.1
Other	18	284068	286236	-2168	-0.8	3852	24.2	2727	29.4
TOTAL	410	5668234	5683844	-15610	-0.3	3624	26.1	3234	35.0

Table 6 shows that the daily assigned volumes on links with traffic counts are less than the observed daily counts by only 0.3 percent. The overall root mean squared error is 35 percent. The high volume roadways tend to be overloaded and the low volume roadways are underestimated. The functional class summary shown in Table 7 confirms this conclusion. The major arterials and above were higher than the counts while minor arterials and below were lower than the counts.

METRO typically uses a two hour PM peak period to validate their regional model. The two hour (3:00 PM to 5:00 PM) PM peak period results are shown in Tables 8 and 9. The assigned volumes are less than the observed counts by 2.8 percent. The root mean squared error is 46 percent. The tables again show that the freeway facilities and major arterials are over assigned and collectors and minor arterials are under assigned.

TABLE 8: PM Peak Period Validation by Volume Level

```
                              Summary Statistics by Volume Level

   Volume Level   Num.      Total      Total Difference      %      Avg.    Avg %     Std.       %
                  Links     Volume     Count                Diff    Error   Error     Dev.     RMSE

       0 to 100     11        844        617        227     36.8       66   117.2       61    156.7
     100 to 250     27       7011       4738       2273     48.0      276   157.1      641    391.3
     250 to 500     66      16700      24202      -7502    -31.0      273    74.6      201     92.4
     500 to 750     74      29328      46650     -17322    -37.1      409    64.8      287     79.0
    750 to 1000     82      43470      71131     -27661    -38.9      557    64.2      289     72.3
   1000 to 2500    266     360431     435307     -74876    -17.2      704    43.0      491     52.4
   2500 to 5000     73     254333     226980      27353     12.1     1055    33.9      817     42.8
   5000 to 7500     24     176885     150426      26459     17.6     1350    21.5     1012     26.7
  7500 to 10000     27     273054     241021      32033     13.3     1393    15.6     1350     21.5
 10000 to 50000     20     218736     220007      -1271     -0.6     1678    15.3     1117     18.2

          TOTAL    670    1380792    1421079     -40287     -2.8      702    33.1      685     46.2
```

TABLE 9: PM Peak Period Validation by Functional Class

```
                            Summary Statistics by Functional Class

   Functional Class   Num.      Total      Total Difference      %      Avg.    Avg %     Std.       %
                      Links     Volume     Count                Diff    Error   Error     Dev.     RMSE

            Freeway     72     650700     597013      53687      9.0     1385    16.7     1191     22.0
 Principal Arterial     42     123101     102690      20411     19.9      799    32.7      522     38.9
     Major Arterial    176     378933     328219      50714     15.5      671    36.0      697     51.8
     Minor Arterial    172     139177     202958     -63781    -31.4      569    48.2      468     62.4
          Collector    190      40775     140943    -100168    -71.1      560    75.5      419     94.2
              Other     18      48106      49256      -1150     -2.3      801    29.3      536     34.9

              TOTAL    670    1380792    1421079     -40287     -2.8      702    33.1      685     46.2
```

6.3 Regional Model Comparison

One of the objectives of the model development was to validate the simulation to a level of accuracy that is equal to or better than the traditional modeling process. Tables 10 through 13 show comparisons of the volumes generated by the METRO highway assignment using the same trip tables used by TRANSIMS to the observed traffic counts. The daily results show that the regional assignment was less than the traffic counts by one percent. This compares to the 0.3 percent in TRANSIMS. The average error, the standard deviation, and the root mean squared error are all better in the METRO assignment than the TRANSIMS simulation.

Where the TRANSIMS simulation showed a tendency to over assign the major facilities and under assign the minor facilities, the METRO assignment results don't show a systematic difference. One interesting observation is that the METRO assignment significantly over loads the low volume facilities. This may reflect an under estimation of intersection delays when volume-capacity relationships are used. On the other hand, the TRANSIMS simulation appears to over emphasize intersection delays or over estimate the performance of freeways.

TABLE 10: METRO Daily Validation by Volume Level

Summary Statistics by Volume Level

Volume Level	Num. Links	Total Volume	Total Count	Difference	% Diff	Avg. Error	Avg % Error	Std. Dev.	% RMSE
0 to 1000	13	20716	7696	13020	169.2	1194	201.7	1451	310.0
1000 to 2500	54	151145	89590	61555	68.7	2021	121.8	1691	158.3
2500 to 5000	85	345951	342307	3644	1.1	1875	46.6	1331	57.0
5000 to 7500	58	280237	361719	-81482	-22.5	2583	41.4	1679	49.3
7500 to 10000	35	272377	304438	-32061	-10.5	1926	22.1	1440	27.5
10000 to 25000	110	1506738	1555812	-49074	-3.2	2393	16.9	2432	24.1
25000 to 50000	19	759962	708268	51694	7.3	4310	11.6	3365	14.5
50000 to 75000	34	2134442	2150614	-16172	-0.8	3622	5.7	2407	6.8
75000 to 100000	2	157939	163400	-5461	-3.3	2730	3.3	1487	3.6
TOTAL	410	5629507	5683844	-54337	-1.0	2378	17.2	2086	22.8

TABLE 11: METRO Daily Validation by Functional Class

Summary Statistics by Functional Class

Functional Class	Num. Links	Total Volume	Total Count	Difference	% Diff	Avg. Error	Avg % Error	Std. Dev.	% RMSE
Freeway	52	2845091	2874830	-29739	-1.0	3913	7.1	3149	9.1
Principal Arterial	24	459029	409706	49323	12.0	2651	15.5	2893	22.7
Major Arterial	104	1111055	1143236	-32181	-2.8	1825	16.6	1392	20.8
Minor Arterial	82	451512	507048	-55536	-11.0	2442	39.5	1619	47.3
Collector	130	431111	462788	-31677	-6.8	1920	53.9	1563	69.4
Other	18	331709	286236	45473	15.9	3785	23.8	2818	29.4
TOTAL	410	5629507	5683844	-54337	-1.0	2378	17.2	2086	22.8

The overall PM peak period comparison is between a positive 4.8 percent difference and a negative 2.8 percent difference. Once again, the average error, standard deviation, and root mean squared error are better in the METRO assignment. The traditional assignment has some trouble with low volume facilities, but in every other respect it is more consistent than the TRANSIMS simulation. This is partially because METRO has been working with and refining the regional network for many years. The ability to control individual link capacities within a traditional assignment model makes it easier to adjust the performance of individual facilities. Much is left to learn about how to code and calibrate a TRANSIMS network to accurately reflect localized conditions. It is also likely that the TRANSIMS results could be improved with additional feedback iterations or adjustments to the Microsimulator calibration parameters.

TABLE 12: METRO PM Peak Period Validation by Volume Level

Summary Statistics by Volume Level

Volume Level	Num. Links	Total Volume	Total Count	Difference	% Diff	Avg. Error	Avg % Error	Std. Dev.	% RMSE
0 to 100	11	2523	617	1906	308.9	187	334.2	281	583.4
100 to 250	27	12515	4738	7777	164.1	339	193.0	510	344.3
250 to 500	66	29347	24202	5145	21.3	269	73.5	260	101.7
500 to 750	74	48798	46650	2148	4.6	293	46.5	277	63.8
750 to 1000	82	70158	71131	-973	-1.4	392	45.2	374	62.3
1000 to 2500	266	428370	435307	-6937	-1.6	453	27.7	377	36.0
2500 to 5000	73	252969	226980	25989	11.4	834	26.8	659	34.1
5000 to 7500	24	166576	150426	16150	10.7	1032	16.5	728	20.0
7500 to 10000	27	252796	241021	11775	4.9	826	9.3	818	12.9
10000 to 50000	20	224955	220007	4948	2.2	1148	10.4	976	13.6
TOTAL	670	1489007	1421079	67928	4.8	499	23.5	521	34.0

TABLE 13: METRO PM Peak Period Validation by Functional Class

Summary Statistics by Functional Class

Functional Class	Num. Links	Total Volume	Total Count	Difference	% Diff	Avg. Error	Avg % Error	Std. Dev.	% RMSE
Freeway	72	625277	597013	28264	4.7	922	11.1	827	14.9
Principal Arterial	42	116790	102690	14100	13.7	686	28.1	461	33.7
Major Arterial	176	363348	328219	35129	10.7	537	28.8	552	41.2
Minor Arterial	172	189718	202958	-13240	-6.5	423	35.8	400	49.3
Collector	190	133852	140943	-7091	-5.0	312	42.1	262	54.9
Other	18	60022	49256	10766	21.9	690	25.2	678	34.9
TOTAL	670	1489007	1421079	67928	4.8	499	23.5	521	34.0

6.4 Tier 1 Screenlines

Model validation at Portland METRO focuses on a subset of screenlines identified as Tier 1 screenlines. These screenlines represent the boundaries between the eight districts defined in the Portland modeling area. METRO compared the simulation results to the daily and two hour PM peak period Tier1 screenline counts. Tables 14 and 15 show the Tier 1 validation statistics for Daily and PM peak period traffic.

Overall, the TRANSIMS volumes are greater than the daily counts on the Tier1 screenlines by 3.3 percent. The largest error is negative 21.6 percent on the I-205 Bridge crossing of the Columbia River. The largest positive error is 19.4 percent in the Western suburbs. The Willamette River bridges in downtown were 8.1 percent high. The majority of screenlines had errors of less than 10 percent.

The traditional assignment volumes are greater than the daily counts by 4.4 percent. The largest error is positive 26.6 percent on the Sellwood Bridge. The largest negative error is 4.1 percent on the I-205 Bridge. The Willamette River bridges in downtown were 5.5 percent high. The majority of screenlines had errors of less than 10 percent. For most of the screenlines, the direction of the error in the TRANSIMS results is the same as the traditional assignment results. Notable exceptions are the Sellwood Bridge and I-5 Hayden screenline. For seven of the twelve screenlines, the TRANSIMS volume was less than the traditional assignment volume. The largest differences were -27.7 percent on the Sellword Bridge and -18.3% on I-205.

TABLE 14: Daily Tier 1 Screenline Statistics

Tier 1 Screenline	Num Links	Daily Counts	TRANSIMS			EMME/2			TRANSIMS-EMME/2	
			Volume	Diff	% Diff	Volume	Diff	% Diff	Diff	% Diff
R-1 St. Johns Br	2	22,610	20,337	(2,273)	-10.1%	22,829	219	1.0%	(2,492)	-10.9%
R-2 Willamette Br	16	464,966	502,649	37,683	8.1%	490,575	25,609	5.5%	12,074	2.5%
R-3 Sellwood Br	2	31,382	28,704	(2,678)	-8.5%	39,715	8,333	26.6%	(11,011)	-27.7%
R-4 Oregon Cty Br	4	109,571	123,308	13,737	12.5%	128,572	19,001	17.3%	(5,264)	-4.1%
R-5 I-5 Br	2	113,838	117,385	3,547	3.1%	127,737	13,899	12.2%	(10,352)	-8.1%
R-6 I-5 Hayden	2	128,300	124,273	(4,027)	-3.1%	134,904	6,604	5.1%	(10,631)	-7.9%
R-7 I-205	2	104,074	81,551	(22,523)	-21.6%	99,780	(4,294)	-4.1%	(18,229)	-18.3%
E-9 W of 122nd	34	348,370	363,304	14,934	4.3%	339,296	(9,074)	-2.6%	24,008	7.1%
E-21 W of Tocoma	20	272,032	280,889	8,857	3.3%	275,566	3,534	1.3%	5,323	1.9%
W-7 Westhills	10	172,226	164,781	(7,445)	-4.3%	178,463	6,237	3.6%	(13,682)	-7.7%
W-9 Tigard Int	6	154,962	155,573	611	0.4%	154,966	4	0.0%	607	0.4%
W-16 NW of Tigard	12	144,000	171,911	27,911	19.4%	164,597	20,597	14.3%	7,314	4.4%
TOTAL	**112**	**2,066,331**	**2,134,665**	**68,334**	**3.3%**	**2,157,000**	**90,669**	**4.4%**	**(22,335)**	**-1.0%**

TABLE 15: PM Peak Period Tier 1 Screenline Statistics

Tier 1 Screenline	Num Links	PM Peak Counts	TRANSIMS			EMME/2			TRANSIMS-EMME/2	
			Volume	Diff	% Diff	Volume	Diff	% Diff	Diff	% Diff
R-1 St. Johns Br	2	3,760	3,191	(569)	-15.1%	3,728	(32)	-0.9%	(537)	-14.4%
R-2 Willamette Br	16	73,358	81,599	8,241	11.2%	85,387	12,029	16.4%	(3,788)	-4.4%
R-3 Sellwood Br	2	5,534	4,513	(1,021)	-18.4%	7,019	1,485	26.8%	(2,506)	-35.7%
R-4 Oregon Cty Br	4	16,160	19,281	3,121	19.3%	22,078	5,918	36.6%	(2,797)	-12.7%
R-5 I-5 Br	2	17,053	18,720	1,667	9.8%	18,563	1,510	8.9%	157	0.8%
R-6 I-5 Hayden	2	15,297	19,831	4,534	29.6%	19,199	3,902	25.5%	632	3.3%
R-7 I-205	2	17,659	12,662	(4,997)	-28.3%	18,110	451	2.6%	(5,448)	-30.1%
E-9 W of 122nd	34	59,835	57,613	(2,222)	-3.7%	59,255	(580)	-1.0%	(1,642)	-2.8%
E-21 W of Tocoma	12	38,221	41,482	3,261	8.5%	40,645	2,424	6.3%	837	2.1%
W-7 Westhills	10	26,726	25,629	(1,097)	-4.1%	30,461	3,735	14.0%	(4,832)	-15.9%
W-16 NW of Tigard	10	28,114	24,403	(3,711)	-13.2%	26,250	(1,864)	-6.6%	(1,847)	-7.0%
TOTAL	**96**	**301,717**	**308,924**	**7,207**	**2.4%**	**330,695**	**28,978**	**9.6%**	**(21,771)**	**-6.6%**

The PM peak period comparison shows greater difference than the Daily results. The TRANSIM results were 2.4 percent high. The results from the traditional assignment were 9.6 percent high. The TRANSIMS differences range from a low of -28.3 percent on I-205 to a high of 29.6 percent on the I-5 Hayden screenline. The average absolute error was 11.2 percent. The traditional assignment differences ranged from a low of -6.6 percent to a high of 36.6 percent. The average absolute error was 11.2 percent. The Willamette River Bridges in downtown were high by 11.2 percent in the TRANSIMS simulation and 16.4 percent in the METRO assignment.

Base on the Tier 1 screenline results and the functional class and volume level results presented in the previous section, it is clear that the TRANSIMS results match the traffic counts reasonably well and are comparable to the METRO assignment. The TRANSIMS results do not appear to be substantially better than the traditional assignment from a regional validation perspective. It is likely that additional work with the TRANSIMS network could improve these results.

6.5 ATR Count Validation

METRO was able to provide detailed traffic count information from six automated traffic recording (ATR) stations. These data made it possible to compare the detailed diurnal distribution of traffic on key facilities to traffic patterns generated by TRANSIMS. The distributions of plan demand and simulated volumes by time of day were compared to the distribution of counts recorded at ATR stations. The ATR data included information for every hour of the day and every day of the year. The hourly counts from weekdays were used to estimate the typical weekday diurnal distribution and traffic volumes. The hourly counts were then smoothed to approximate 15-minute counts.

Figures 48 through 52 compare the 15-minute counts, Router demand and simulated volumes at several ATR stations. The difference between the Router demand and the simulated volume shows how the Microsimulator spreads the traffic to later time periods due to capacity constraints. The Router demands are generally higher than the ATR counts. Since all of the ATR counts are on freeways, this suggests that the simulated speeds on the freeways are too high or the speeds on the arterials are too low. Simulated volumes during the peak periods and the middle of the day are slightly higher than the observed counts. The simulation is lower than the counts during the early morning and evening hours. This indicates that retiming the activities to have a slightly greater proportion of trips starting very early or late in the day could decrease the overloading that occurs during peak and mid-day hours and improve the overall results. In general, the simulation results replicate the observed traffic volumes reasonably well.

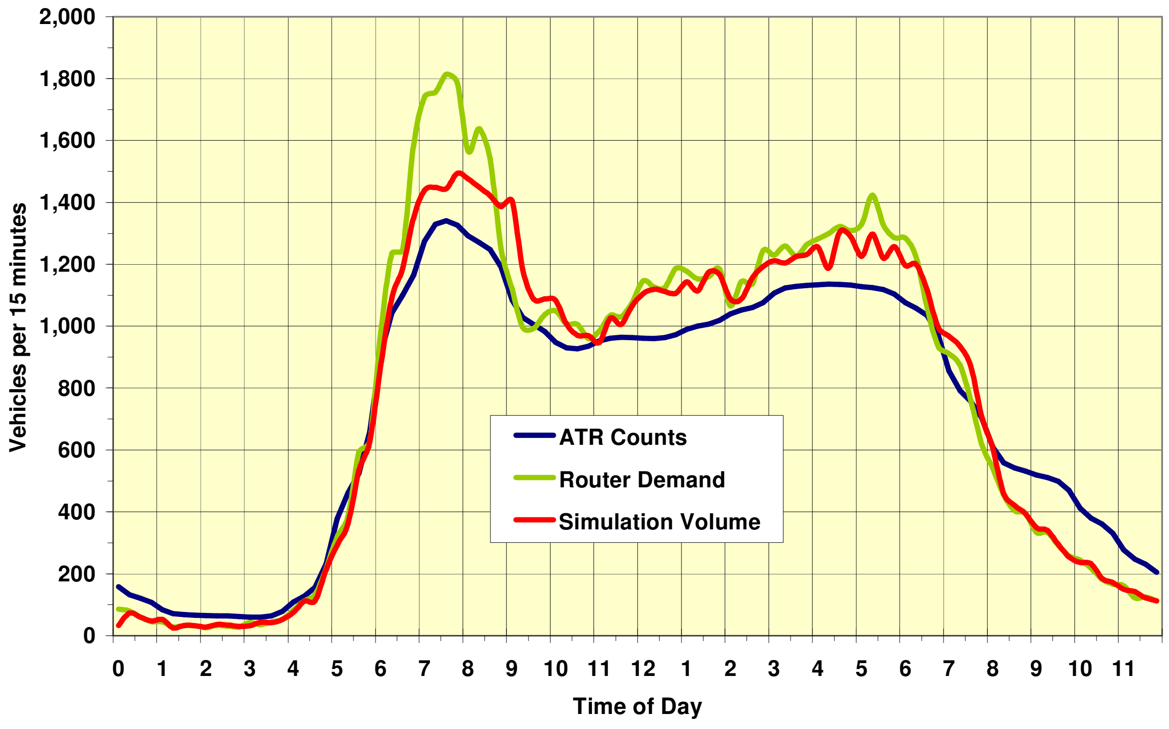


FIGURE 48: US 26 – Vista Ridge Tunnel – East Bound

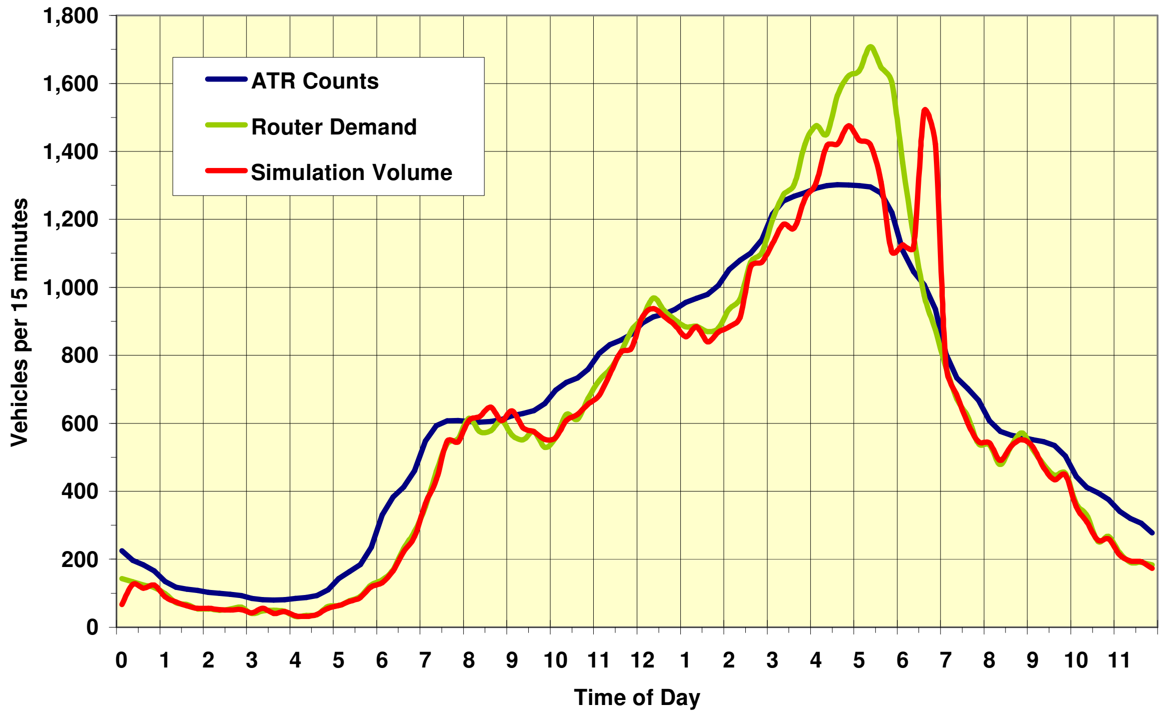


FIGURE 49: I-5 – Interstate Bridge – North Bound

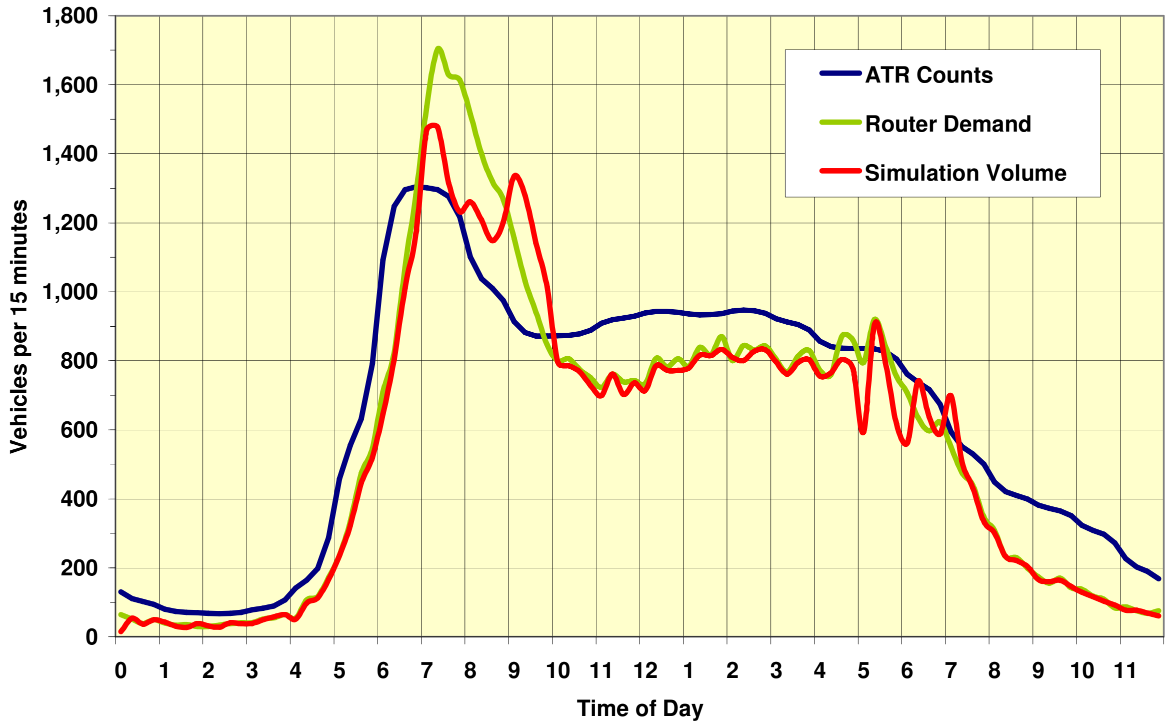


FIGURE 50: I-5 – Interstate Bridge – South Bound

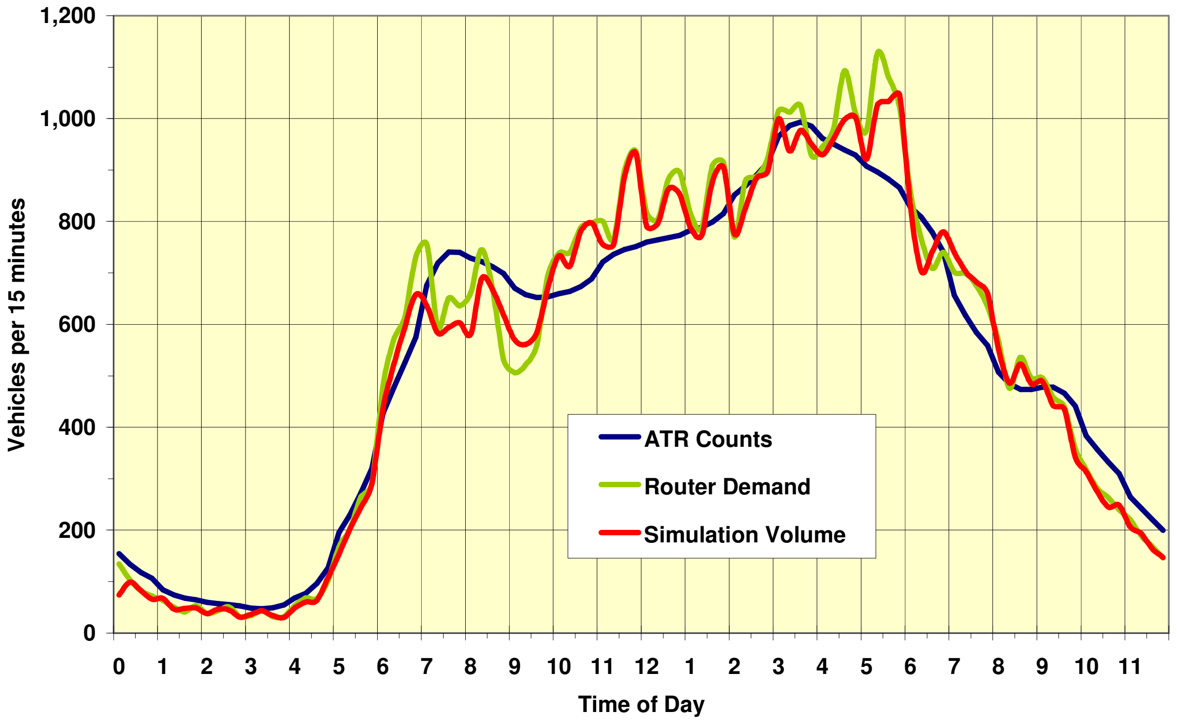


FIGURE 51: I-405 – Stadium Freeway – South Bound

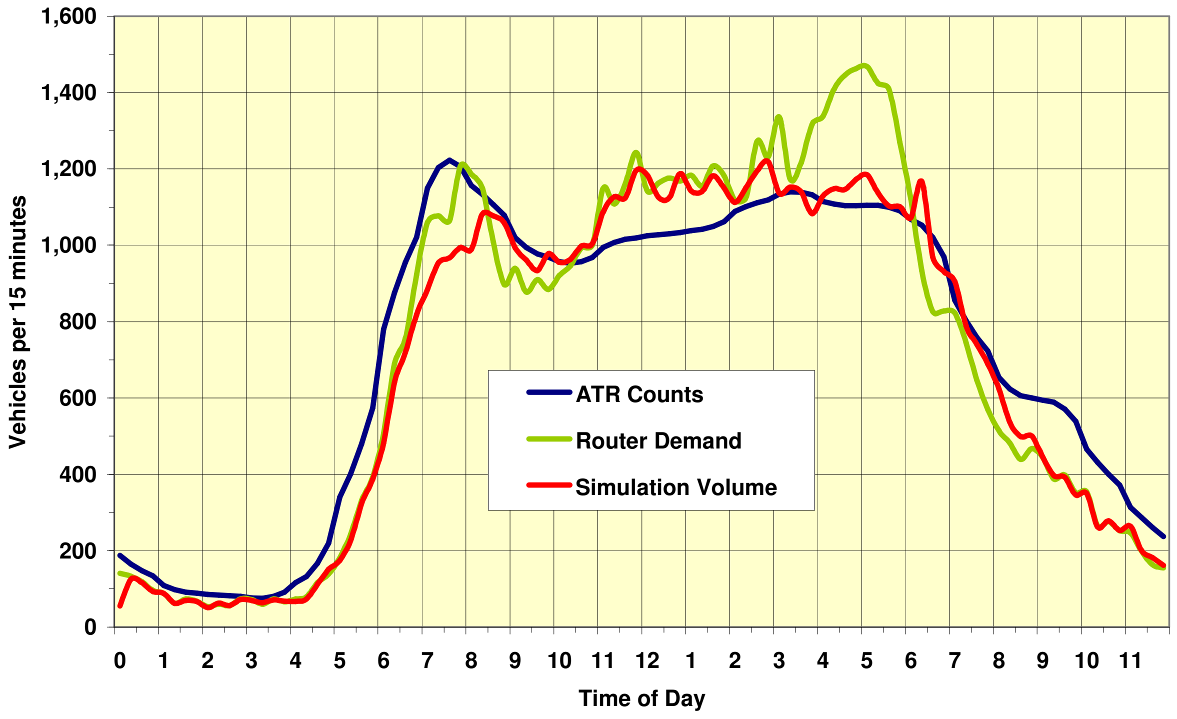


FIGURE 52: I-5 – Marquam Bridge – North Bound

6.6 Speed Data

The LinkSum program was used to summarize Microsimulator data to generate 15-minute link travel times on some major freeway and arterial facilities. Figures 53 and 54 show the diurnal distribution of the 15-minute average travel speeds on US-26 Vista Ridge tunnel and I-84. Both plots show drops in the average travel speeds during morning and the evening peak periods (in the peak direction of travel) as expected. The speed fluctuations on I-84 at Hoyt are significantly more erratic than on US-26 at Vista Ridge. This is not totally unexpected, but it is also not easily explained. I-84 is frequently congested and does breakdown on a regular basis. This was also one of the more difficult areas to stabilize. The tradeoffs between using the freeway and the local arterial network were very sensitive to small changes in the network characteristics. An improper balance often resulted in cascading queues. These results suggest that additional refinement in the travel paths in this area may be warranted.

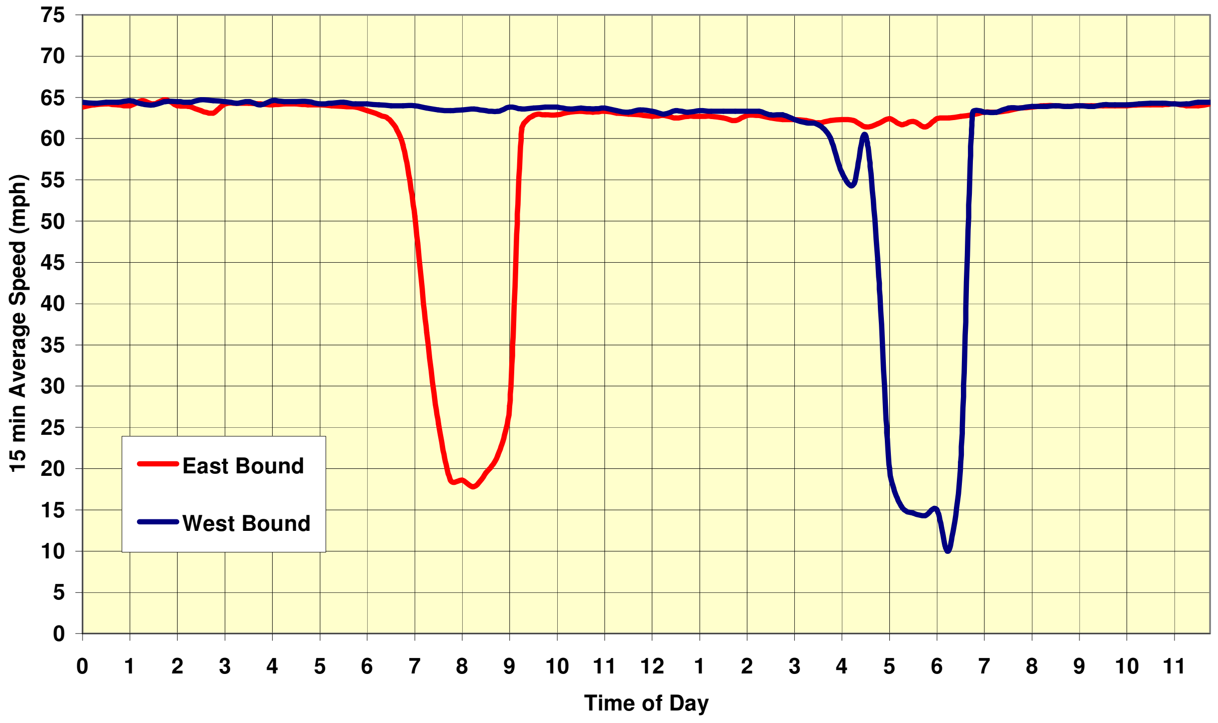


FIGURE 53: US – 26 Vista Ridge Speeds

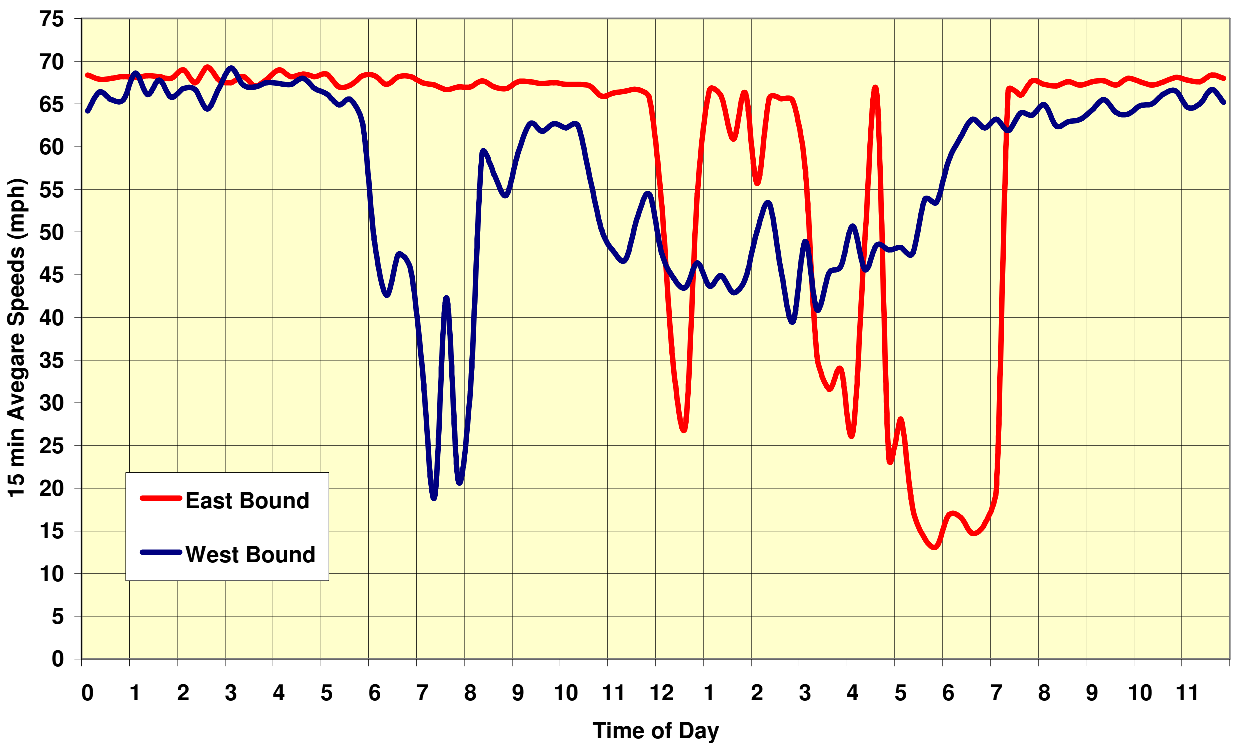


FIGURE 54: I – 84 Hoyt Speeds

Actual speed data for the facilities shown in Figures 53 and 54 were not available. The evaluation was limited to local experience. METRO had data for a number of arterial segment travel time studies. These studies included a number of moving car runs during specified time periods. The simulation results were summarized in a similar way. This analysis showed that the TRANSIMS speeds were higher than the moving car results. METRO re-evaluated the speeds used in the TRANSIMS network and found that downward adjustments were warranted. Most of the links in the City of Portland were reduced by one cell per second speed increment (i.e., 7.5 meters / second or about 16 mph). A number of Microsimulator runs were made based on these changes. The speed estimates improved, but still were not totally satisfactory. Additional tests using different acceleration and deceleration parameters are underway in an attempt to further improve the speed estimates.

Chapter 7 Sensitivity Testing

Chapter 6 showed that the TRANSIMS model replicated the observed traffic counts, diurnal distributions, and speeds reasonably well. In addition to basic network performance, the model application process was able to approximate user-equilibrium for each traveler as well. In other words, 98 percent of the travelers could not reduce their travel time by changing paths. This chapter documents the sensitivity of the network performance and the user-equilibrium condition to changes in supply and demand. The sensitivity tests were designed to understand the behavior of the model so that it can be confidently used for future year forecasting and alternatives analysis.

7.1 Test Scenarios

The test scenarios were designed to evaluate the impact of network changes and travel demand on the performance of the system and the TRANSIMS modeling process. Four tests were implemented as part of this study. They include:

1. Traffic operations/policy changes
2. Increase in Demand
3. Removal of the Hawthorne Bridge
4. Removal of the Hawthorne and Morrison Bridges

The first test was designed to quantify and analyze the impacts of implementing improvements to traffic operations. For this test, a regional change to traffic signal plans was implemented. The second test studied the impacts of loading the year 2000 trip tables to the 1994 network. The third test studied the impacts of closing the Hawthorne Bridge in downtown Portland. This test was designed to study how the regional simulation responds to a small reduction in network supply. The final test studied the impact of closing both the Hawthorne and the Morrison bridges in downtown. This test was designed to study the impacts of major reduction in network capacity. The results of each of these tests are presented in the sections that follow.

7.2 Traffic Operational Changes

Since the TRANSIMS Microsimulator considers intersection traffic controls and other traffic operational conditions, a test scenario was constructed to identify how the simulation results would change if the operating plan changed. The existing network included fixed timed signals with signal progression in downtown Portland and a few other locations. For this test, all of the demand actuated signals along major highways and arterials in the Portland area were changed to fixed timed signals with signal progression. Figure 55 shows the signal operating plan included in the original network. For this test, most of the major intersection signals were changed from demand actuated operating plans to a fixed timed signal system with signal progression. Figure 56 shows the signal operating plan included in this test. The TRANSIMS feedback process was applied to the test network until the network performance stabilized and user-equilibrium was reestablished.

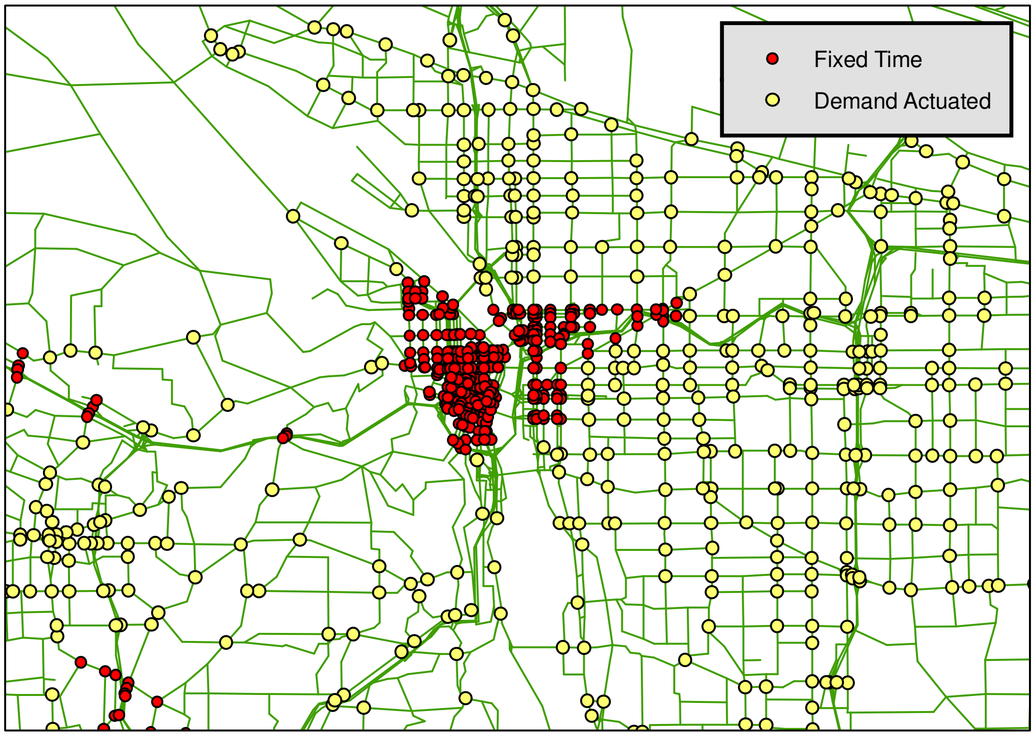


FIGURE 55: Original Signal Operating Plan

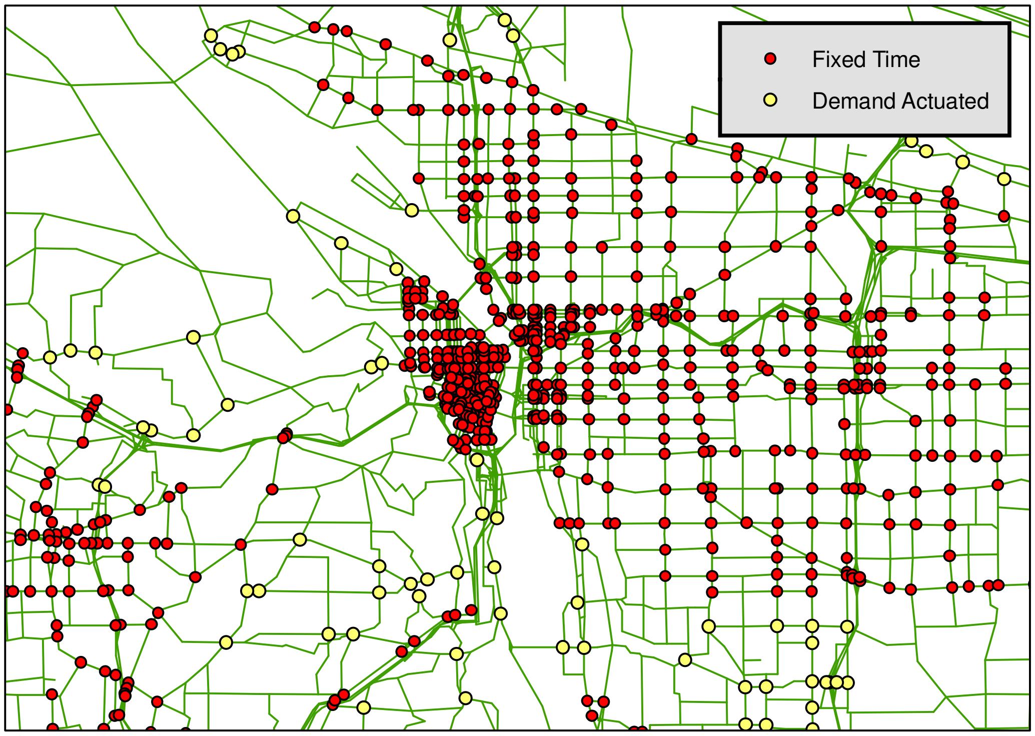


FIGURE 56: Signal Operations Test Scenario

The traffic operational test started by loading the final plan set from the original stabilization process to the new network configuration. The initial simulation showed increased congestion during the morning peak period. In other words, the new signal timing plans changed the capacity constraints and performance at a number of intersections. These changes resulted in traffic congestion and queues at different places within the network. Several time-based stabilization and equilibrium convergence re-routes were needed to enable the travel plans to adjust to the new conditions.

Figure 57 shows the impact of the traffic operational change on the average travel speed by facility type. The chart shows that changing the traffic signals on arterials from demand actuated to fixed timed signals increased the average travel speeds by 0.4 mph. The fixed timed signals with signal progression increased the throughput on the major and minor arterials by minimizing the number of times the vehicles stopped at the intersections. These impacts are relatively small and may simply represent the stochastic effects of the Router and Microsimulator on the assigned trips. Notwithstanding, the ability of the TRANSIMS process to consider traffic operational changes is well beyond the capabilities of traditional modeling methods.

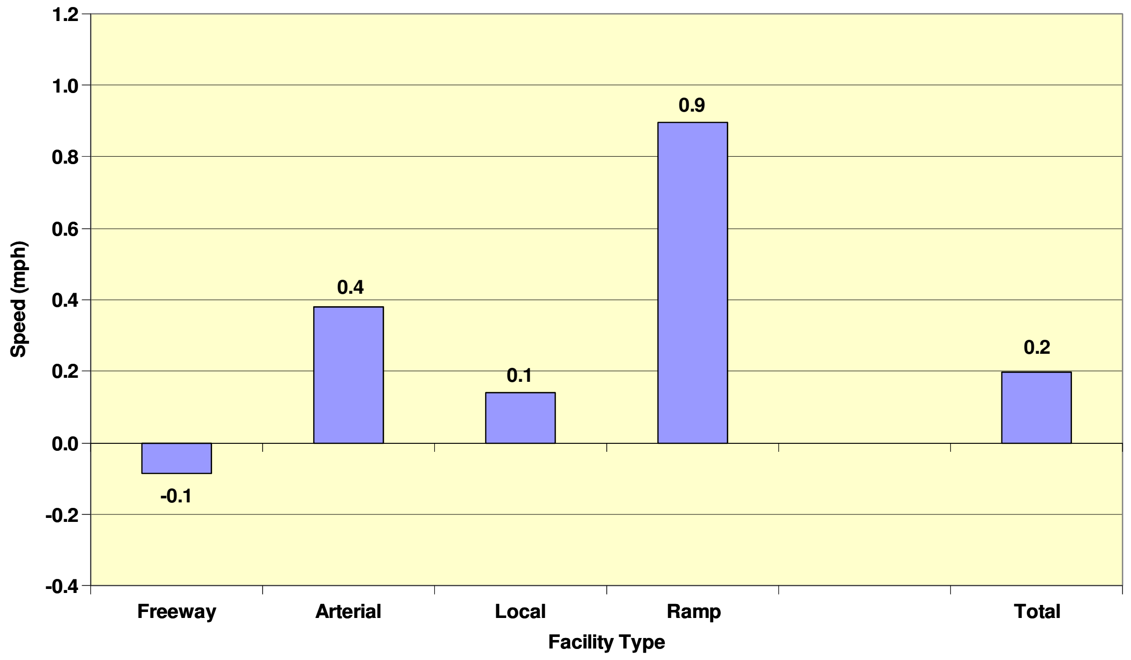


FIGURE 57: Change in Average Speeds by Facility Type

7.3 *Increase in Demand*

One concern about using the TRANSIMS model for future year applications is that travel demand cannot exceed the network throughput. When demand exceeds the capacity of the network, cascading queues form that often result in system gridlock. Under these conditions, the Microsimulator may not be able to satisfy all of the demand or achieve user-equilibrium convergence. This makes it difficult or impossible to use the TRANSIMS model for testing future year alternatives where travel demand increases significantly, but the network is basically unchanged (e.g., No-Build). The best that TRANSIMS can do is identify the percentage of demand that can be accommodated by the network. This may be a very useful performance measure, but it is significantly different from current planning practices.

In order to determine how the TRANSIMS model responds to a significant increase in demand, the regional trip tables for the year 2000 were loaded onto the 1994 network. This represented a 14.9 percent increase in travel demand with no change to the network. The increased demand was loaded onto the network using the full assignment process. The process was able to accommodate the increased demand and generate logical results. The network was noticeably more congested. The number of trips increased by 14.9 percent, but the number of lost vehicles increased by 57.5 percent. The lost vehicles were 1.8 percent in the original 1994 application. In the sensitivity test, 2.6 percent of the vehicles loaded onto the network did not complete their trip.

If the demand could not be accommodated, traffic operational adjustments (e.g., signal timing or phasing plans) or travel schedule changes (e.g., peak spreading) could be considered. Changes such as these would be logical in the context of developing a realistic balance between supply and demand. On the other hand, the comparison of alternatives is more complex. Rules and procedures for defining what can and cannot be done in the context of an alternatives analysis need to be defined.

7.4 Removing the Hawthorne Bridge

The sensitivity test to study the impacts of a minor reduction in supply involved removing the Hawthorne Bridge from the base network. The Hawthorne Bridge is one of several bridges crossing the Willamette River in downtown Portland. It carries about 6 percent of the daily river crossings. Figure 58 shows the location of the Hawthorne Bridge in relation to downtown and the other river crossings.

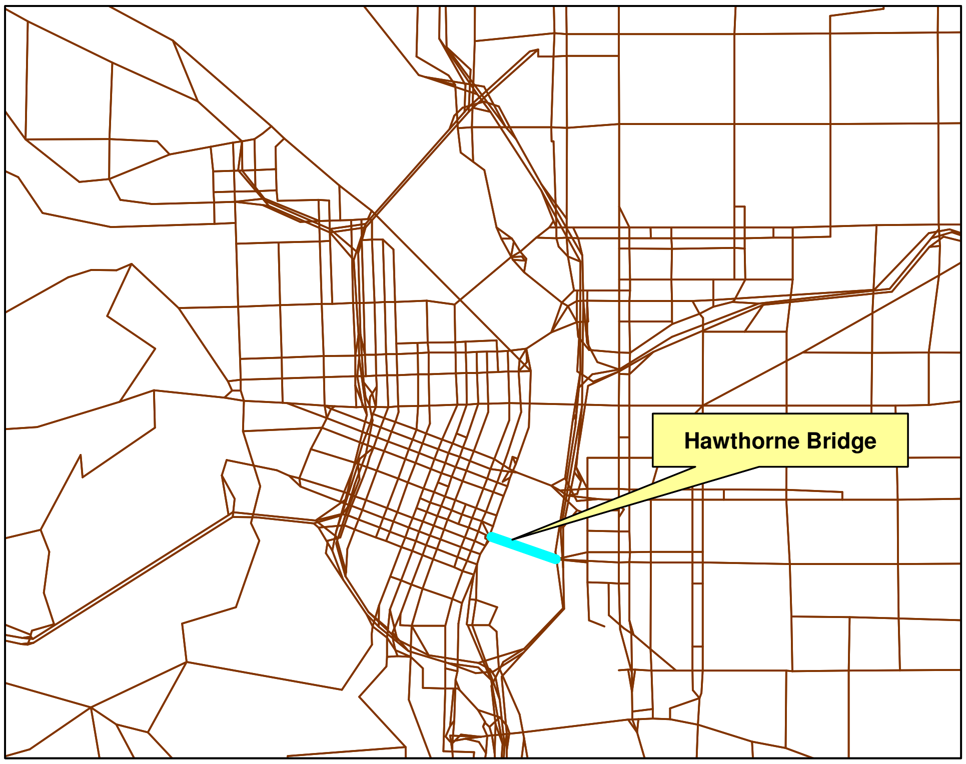


FIGURE 58: Location of the Hawthorne Bridge

The PlanSelect program was used to identify the travel plans in the original plan file that utilized the Hawthorne Bridge for their trip. These travelers were re-routed using the modified network and the Microsimulator was used to re-estimate the link travel times. Only a few travel time stabilization and equilibrium convergence runs were required to bring the system back into balance. Almost all of the changes in volume and speed were in the immediate vicinity of the bridge. This means that the process was able to effectively identify the network change and focus travel plan adjustments to a relatively narrow impact area. This was encouraging since the equilibrium assignment procedures used in traditional models have a reputation for generating noticeable changes in volumes far away from insignificant changes in the network coding.

7.5 Removing the Hawthorne and Morrison Bridges

Both the Hawthorne and Morrison bridges in downtown Portland were closed to test a major reduction in network supply. As seen in Figure 59, these bridges are next to each other and provide a critical link between East Portland and the heart of downtown. Together they carry 15 percent of the daily river crossings into downtown Portland.

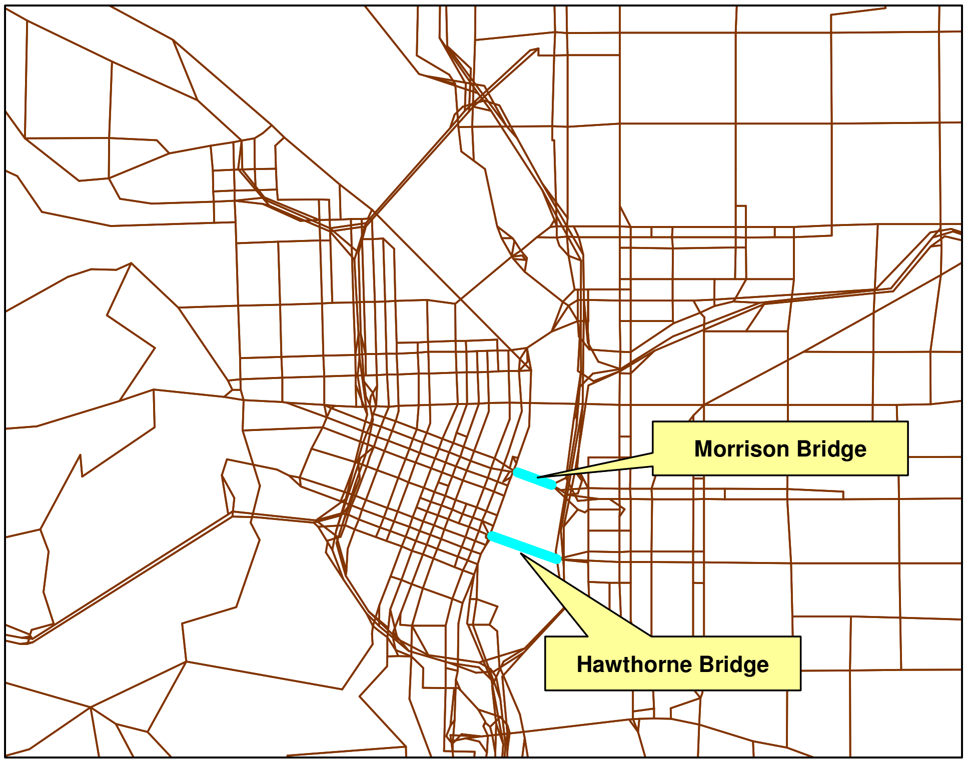


FIGURE 59: Locations of the Hawthorne and Morrison Bridges

Since the network change was fairly significant the decision about how to perform the analysis was not as straightforward. It could be implemented in the same way as the Hawthorne Bridge analysis: by pivoting off of the final plan set and simulation results from the original application. Alternatively, the whole assignment process could be run from scratch. It was decided that both methods would be applied and the results compared to gain a better understanding of the overall impact of the assignment method on the estimated volumes.

7.5.1 Pivot Method

The pivot method started the sensitivity analysis from the final plan set and simulated speeds of the original application. The PlanSelect program was used to identify the travelers who used the Hawthorne or Morrison bridges to complete their trip. These travelers were re-routed on the modified network using the link delays from the original simulation. The resulting plans were merged with the original plan set and the Microsimulator was applied. The simulation was highly congested in the vicinity of the bridge closings. The resulting link delays were used to re-route a random 15 percent of the trips. This was followed by several stabilization runs that re-routed 10 percent of the trips each time. It took 12 stabilization runs to eliminate the congestion impacts.

Figure 60 highlights the links were the volumes changed by plus or minus ten percent or more as a result of the bridge closures. The results show logical increases and decreases in traffic volume in and around the bridge closures in downtown. Most of the impacts are within a mile of the network change. There are, however, a several places farther away from the impact area that changed as well. These changes are not likely to be a direct impact of the bridge closers. They reflect areas within the region were the simulation was not fully stabilized. By randomly selecting 10 percent of the trips for re-routing, the Router was provided with an opportunity to further refine the path choices in these areas. If the user wished to avoid these impacts and focus exclusively on the travelers impacted by the bridge closures, a different selection method would need to be used. This method would limit the travelers selected for re-routing to the traffic on the bridge links plus the links that changed as a result of re-routing the bridge traffic.

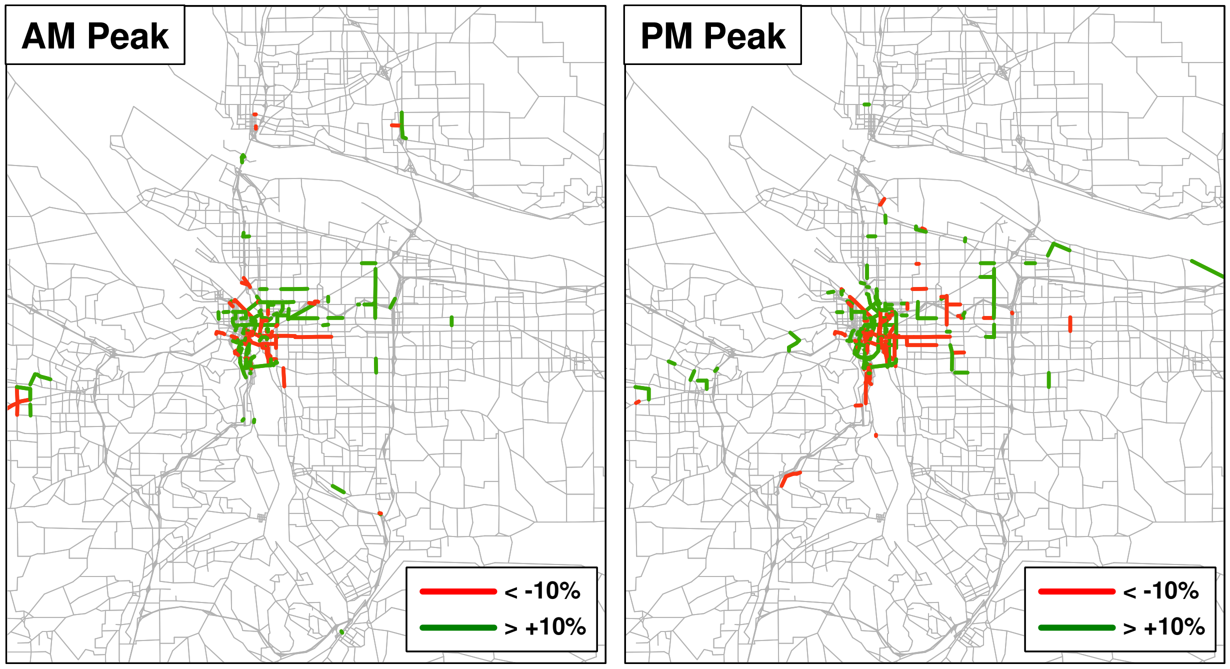


FIGURE 60: Volume Changes from the Pivot Method

7.5.2 Complete Re-Run Method

In addition to the pivot method, a complete re-run of the assignment process was performed using the modified network. This process starts with free flow speeds on the links, incrementally loads the trips using V/C-based volume-delay functions, and then a full set of stabilization loops. This process took longer to run and generated different volumes. The volumes were not, however, illogical. Table 16 shows a comparison of the original Willamette River bridge crossings to the new estimates. The differences in the assigned volumes look reasonable. The increase in volumes on the Burnside, Steel and Marquam Bridges are logical diversions to neighboring bridges. The overall drop in river crossings is 1.8 percent. This includes about 1,000 trips that originally made two bridge crossings that no longer cross the river at all.

TABLE 16: Sensitivity Test – Removal of Hawthorne and Morrison Bridges

Bridge Name	Base Volume	No Bridges Volume	Difference
Fremont Bridge NE	54,733	55,644	911
Fremont Bridge SW	63,545	65,678	2,133
Broadway Bridge NE	13,929	17,323	3,394
Broadway Bridge SW	11,894	14,396	2,502
Steel Bridge NE	7,068	16,497	9,429
Steel Bridge SW	4,388	7,788	3,400
Burnside Bridge NE	17,047	24,347	7,300
Burnside Bridge SW	11,947	25,376	13,429
Morrison Bridge NE	27,132	-	(27,132)
Morrison Bridge SW	30,808	-	(30,808)
Hawthorne Bridge NE	13,932	-	(13,932)
Hawthorne Bridge SW	12,051	-	(12,051)
Maquam Bridge NE	74,869	81,755	6,886
Maquam Bridge SW	59,002	64,674	5,672
Ross Island Bridge NE	33,039	41,677	8,638
Ross Island Bridge SW	41,161	50,997	9,836
Sellwood Bridge NE	19,371	19,749	378
Sellwood Bridge SW	20,391	21,229	838
Totals	**516,307**	**507,130**	**(9,177)**

Figure 61 shows the percentage change in peak period volumes as a result of the complete re-run. Most of the links in downtown show some impact from the bridge closers. These results, however, show greater impacts throughout the region. A number of the changes are on feeder arterials to freeways that pass through downtown. Some of these long distance trips appear to be avoiding the increased congestion on I-5 in downtown by using the I-205 by-pass. Others appear to be random, unrelated changes. These changes may not be as significant as a traditional equilibrium assignment model, but they raise the same types of issues and concerns.

These results raise the question, when is it appropriate to implement a complete model run versus pivoting off of the base line forecast? Pivoting makes sense for small network changes, but can become considerably complex for major network changes. This question could also be extended to future trip tables as well. Is it better to add the change in trips to the base year plan set and simulation results, or generate totally new activity patterns based on changing network conditions? TRANSIMS offers a world of possibilities that the industry has only begun to explore.

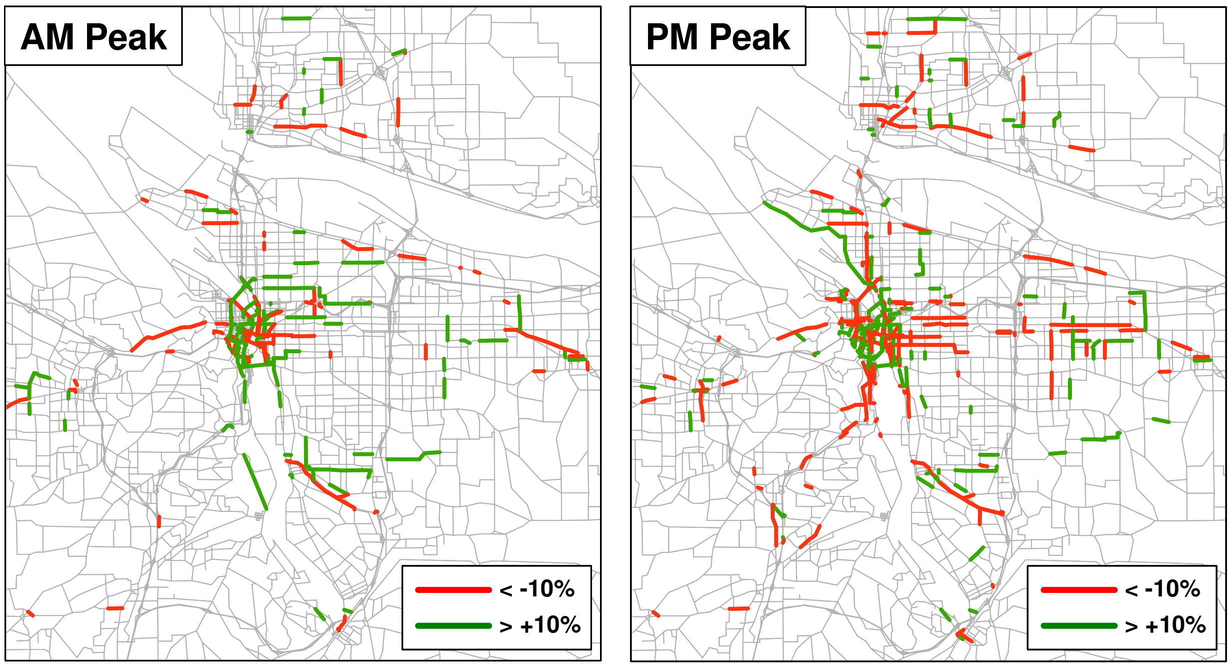


FIGURE 61: Volume Changes from the Complete Re-Run Method