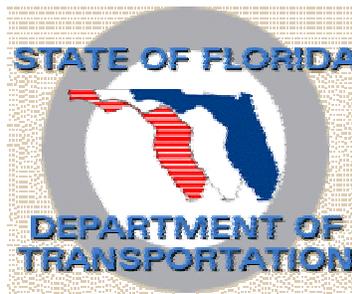


OPERATIONAL TESTING OF TWO INTERIM PRODUCTS FROM THE TRAVEL MODEL IMPROVEMENT PROGRAM Final Report

Research Center
Florida Department of Transportation
605 Suwannee Street, MS 30
Tallahassee, FL 32399-0450



Compiled by

Ram M. Pendyala, Ph.D.

Associate Professor & Principal Investigator

Department of Civil and Environmental Engineering

University of South Florida, Tampa, FL 33620

Documentation prepared by

Los Alamos National Laboratory

Los Alamos, New Mexico 97545

Contract No. BC009
2002

Acknowledgements

Special thanks go to the project manager, Mr. Robert G. McCullough, P.E., with the Systems Planning Office of the Florida Department of Transportation, for his technical guidance and advice through the course of the project. The project team is grateful to the Los Alamos National Laboratory (LANL) TRANSIMS development team and the PriceWaterhouseCoopers (PWC) TRANSIMS-DOT development team for their assistance in carrying out the case study tests and for providing documentation that has been incorporated in this report.

Disclaimer

The contents of this report do not necessarily reflect the official views or policies of the Florida Department of Transportation or its constituent divisions. This report does not constitute a standard, specification, or regulation.

Table of Contents

PART 1: STUDY OVERVIEW	1-1
PART 2: PARAMETERS AND INPUT DATA	
Chapter 1: Network	2-1-1
Chapter 2: Population.....	2-2-1
Chapter 3: Activity Generation.....	2-3-1
Chapter 4: Router.....	2-4-1
Chapter 5: Microsimulation.....	2-5-1
PART 3: GENERAL RESULTS	
Chapter 1: Population and Activities.....	3-1-1
Chapter 2: Routes	3-2-1
Chapter 3: Microsimulation.....	3-3-1

Part 1

Study Overview

Study Overview

1. INTRODUCTION AND PURPOSE OF THE PORTLAND STUDY

This document provides a high-level overview of the features of the TRANSIMS software package that were used in the Portland Case Study. As TRANSIMS-DOT was not completed during the period covered by this project, the project team had to perform the case study in collaboration with Los Alamos National Laboratory and Portland Metro using the Portland area databases.

TRANSIMS is not a model. It is a modeling system that permits the modeler to keep track of and modify the behavior of each individual in a synthetic population. The TRANSIMS technology consists of software modules and file protocols that make this modeling possible.

The purpose of the Portland Study is not to develop a complete calibrated model of 1996 Portland. Rather, it allowed the developers of the system to understand transportation planning methods and the models and data that support these methods. Additionally, there are three demonstration purposes of this study:

- 1) To show that the software runs for a regional problem (Portland, with every street represented) on available computers in a reasonable amount of time (less than 24 hours for each module).
- 2) To show that TRANSIMS can be made responsive to the variables and conditions that planning organizations deem to be important, and that models produced by TRANSIMS may be calibrated for use in a forecast setting. For example, are modules responsive to the variables that influence mode choice such as downtown parking costs, availability of transit service, and household income? Can TRANSIMS be made to match screen line counts such as bridge crossings in the base case year, and can the “validation constants” obtained for these matches be used in a forecast setting? These calibrations use meta-methods and methods that are external to TRANSIMS and are not part of the core TRANSIMS technologies.
- 3) To show that it is possible to develop TRANSIMS meta-methods and methods that would allow matches to Portland traffic and transit counts for a base year. In transportation planning exercises, it is critical that the base year be calibrated to existing conditions. In this particular Study, however, the actual calibration is less important than demonstrating working and responsive software and methodologies. There are two reasons for this. First, as in any planning study, it is time consuming and expensive to determine the calibration constants to validate the model for the base year. The limited resources for this study were better utilized in the development of calibration methods rather than fine tuning the models and the input data. Second, a calibration or validation of a base year depends on the metropolitan region

being studied, the base year scenario, and the input data such as the network transit schedules and land-use. The “calibration” constants derived in this study will have to be regenerated for other studies in other metropolitan regions and even studies for Portland Metro. The main goal of this exercise is to develop methodologies that permit this calibration and communicate the techniques that would accomplish this goal. If items 1 and 2 above are met, then it is obvious that traffic and transit counts could be matched using the TRANSIMS software modules and methodologies similar to the ones shown here.

Throughout this set of documents it is apparent that the methodology development to choose modes and match screen-line counts is in a testing stage. In some cases, bridge crossing screen lines for example, the first methods developed do not work entirely as expected. These preliminary methods do, however, show that TRANSIMS can be made to reset the resulting activity set such that screen line or mode splits approach the actual counts and, hence, satisfy item 2 above. In critical cases, such as too many vehicles crossing bridges and causing unrealistic traffic jams, the original methods were modified to produce the desired results. In some of these cases, rather than start over, methodologies were developed to refine the results of the original methodology. To save time and resources where the calibration could be improved in non-critical cases, the original methods were not modified if it was clear that only a slight change in methodology was needed.

The complete set of Portland Study Reports documentation comprises eight volumes that are available at the TRANSIMS website: <http://transims.tsasa.lanl.gov>. The purpose of the various volumes is to document the procedures and methodologies used in the Study and to give guidance to the development of new methodologies where the need arises. At this time, volumes six and seven are not yet available as the Portland Case Study is not yet complete.

In addition to an introductory volume, the Portland Study Reports consists of the following volumes:

Volume Two *Study Setup: Parameters and Input Data*

The input parameters (called configuration file keys) used to run each of the TRANSIMS modules are described. It outlines the data used to construct the TRANSIMS network. The external inputs to the various modules from external sources are also described.

Volume Three *Feedback Loops*

The basic TRANSIMS philosophy is to do minimal modeling, requiring as little data as possible to start the process. Feedback loops are critical, since they, rather than an econometric model, are the main modeling tools in TRANSIMS. Volume Three describes feedback methodologies to restrict the number of vehicles crossing the bridges in Portland, feedback to stabilize the traffic patterns, and feedback for mode choice.

Volume Four *General Results (not validation)*

As each TRANSIMS module is executed, an output file is generated. Volume Four contains information about each data set produced during the Study. This

includes the file sizes, the computer used, the time it took to generate the file, and the external data used by the modules. A description of any transformation made to this data for analysis during the Study is also included.

Volume Five *Environmental Results*

Emissions estimates from the final microsimulation run are presented.

Volume Six *Comparisons to Portland Data*

The main purpose of this study is one of calibration methodology development rather than complete validation of a base case year. Volume Six was written by Portland Metro, and it characterizes a typical validation methodology for transportation modeling. It contains comparisons of the actual ground counts, mode splits, and trip length distributions with the output of the model. In those cases where the matches could be improved, possible methodologies for making such improvements are given in Volume Seven.

Volume Seven *Summary and Forecast Methods*

The results of the Study are summarized in Volume Seven. The validation is discussed, and in those cases where the validation numbers could be improved, suggestions of the revised methodologies are given. The Study itself is for a base year. Forecast methodologies using the calibration constants determined from a base case study are presented.

Volume Eight *Appendix: Scripts, Configuration Files, Special Travel Time Functions*

TRANSIMS has a flexible Framework that allows for the construction of various models of transportation systems. Some of these systems are constructed by using information from a data set from one TRANSIMS module to change some of the input conditions of a different (or the same) module. A calibrated model is built from the basic TRANSIMS building blocks in this way. Volume Eight contains the scripts, configuration files, and special functions that were applied to produce the final results of this study.

The Los Alamos National Laboratory TRANSIMS development team is working closely with the Portland Metro modeling staff to perform comparisons with existing Portland data and conduct a review of the case study. The project team will continue to work with these entities in the future to examine the potential of applying TRANSIMS technologies in Florida.

2. THE INPUT DATA

While collating data collected for other purposes to make it compatible with TRANSIMS was a tremendous chore, only existing data were used in this study. No new data collection efforts were undertaken by Portland Metro to support the Study. Some of the data supplied by Metro, such as zone-to-zone travel time tables, could have been generated from TRANSIMS results. Other data, such as the activity survey data, is basic input to models of travel behavior other than TRANSIMS.

The most time consuming data preparation for this study was the development of a roadway network that includes

- all local streets,
- land-use characteristics on a block face,
- lane connectivity at intersections,
- intersection signalization, and
- transit schedules, routes, and stops.

The daunting task of constructing the detailed TRANSIMS network for the Portland Study was made simpler by using automated procedures to generate the lane connectivity. Additionally, generic traffic signals were used throughout the study. Full detail of the network generation is given in Volume Two (*Study Setup: Parameters and Input Data*), Chapter One (*Network*) of the Portland Study reports. It should be noted that the commercial version of TRANSIMS will contain a network editor that will ease the task of network construction.

Portland Metro's two-day activity survey was used extensively in this study. Data from the first day of the survey for 3,473 households, out of the 5,863 households in the complete survey, were used to support the TRANSIMS Activity Generator. This use is described in Volume Two (*Study Setup: Parameters and Input Data*), Chapter Three (*Activity Generation*) of the Portland Study Report series. Some calibration target values, such as the number of work tours that cross the Columbia River, were estimated by Portland Metro using the entire survey data set.

Itinerant traveler and truck trips were generated using trip tables supplied by Portland Metro. This is described in Volume Two (*Study Setup: Parameters and Input Data*), Chapter Three (*Activity Generation*) of the Portland Study Report series.

Zone-to-zone travel times by mode and time of day could have been generated using the results of a TRANSIMS analysis. Here, however, existing zone-to-zone travel time tables were used. More information is in Volume Two (*Study Setup: Parameters and Input Data*), Chapter 3 (*Activity Generation*) and Volume Three (*Feedback Loops*), Chapter Two (*Calibration of River Crossing Screen Lines*) of the Portland Study Report documents.

3. THE MODULES

TRANSIMS consists of a series of building blocks, or modules, that produce populations, activities for the populations, routes for travelers, and microsimulated traffic dynamics. This Framework is shown in Fig. 1, which shows each of the modules and a series of arrows. The arrows give the possible pathways to transfer data from one module to another. The TRANSIMS Framework allows these modules to be executed in any desired order by a set of scripts. As each module in the Framework is executed with scripts, selected data is collected in an iteration database to be used by the other modules. This allows for information from, say, the microsimulated traffic dynamics to be fed back to the Route Planner to produce a new set of more realistic routes for selected travelers. Feedback of information from one module to another is important in TRANSIMS because it allows the results to be modified and fine tuned to produce realistic traffic and transit dynamics in the system. The feedback possibilities are shown by the red arrows in Fig. 1. The use of these feedback pathways is discussed in the next chapter of this document. Feed-forward data transfers are shown by the black arrows. A complete description of the algorithms and these pathways is available on the TRANSIMS web site.

A major TRANSIMS technical feature is that the identity of individual synthetic travelers is maintained throughout the entire simulation and analysis architecture. All synthetic travelers are generated as part of the development of a synthetic population for a specific metropolitan region using a variety of data sources including census, surveys, etc. Activity times and locations are computed for each individual. The intermodal route plans generated by the Route Planner maintain individual identities, as does the microsimulation. The resulting simulation output can provide a detailed, second-by-second history of every traveler in the system over a 24-hour day. A variety of impact analyses can be conducted using these results. This approach produces a simple, consistent architecture—one that provides planners with deeper insight into the underlying, second-by-second dynamics of the traffic system under different local (e.g., traffic signals) and global (e.g., congestion) conditions.

Fig. 1 illustrates that the various modules of TRANSIMS may be called in any order with different inputs. TRANSIMS itself is not a model of a transportation system. Rather, TRANSIMS is a modeling system. Models are built using TRANSIMS by determining the order of the calls to the modules, by the information that is allowed to flow from one module to another, and by the meta-methods and methods external to TRANSIMS that manipulate these data to build models.

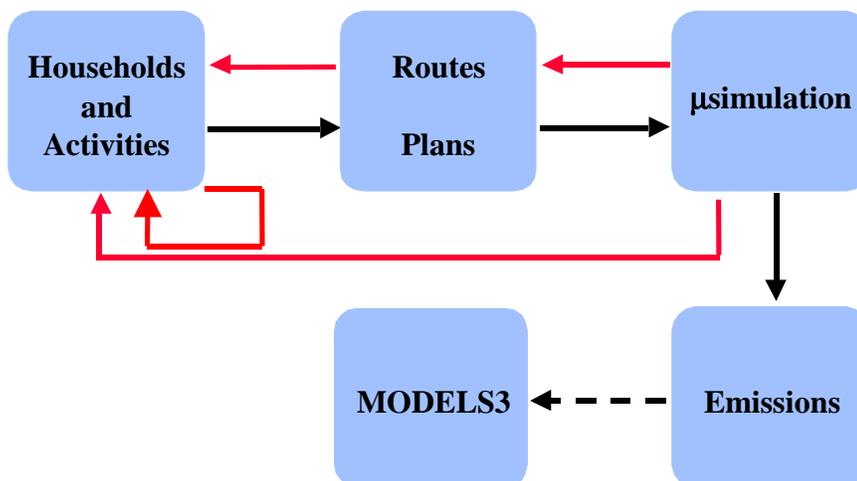


Fig. 1. The highest level view of TRANSIMS consists of four major modules. The modules are household and activity generation, intermodal router, traffic/mobility microsimulation, and an emissions estimator. Feedback based on the modules is used to re-plan and modify activity demand and as a modeling tool. Multiple analyses, in addition to air quality analyses, can be performed using microsimulation results.

4. THE ROLE OF FEEDBACK

Feedback and controlling feedback are important functions of the Framework. One of the functions of feedback is to obtain stability in the results. As in assignment methodologies, the Route Planner may place more vehicles on links than the capacity of the link may allow. This may cause congestion to spill back onto other links. Results of the microsimulation in these cases can be fed back to reroute selected travelers to stabilize this situation. The other function of feedback is to model various aspects of transportation systems such as controlling traffic movement across the Columbia River.

In the Study, feedback was used in many capacities. Three of the important ones are:

- 1) First, as mentioned above, feedback was utilized to stabilize the traffic. The feedback pathways to accomplish this are depicted in Fig. 1 by the black arrow from the Route Planner to the microsimulation and the feedback red arrow in the reverse direction.
- 2) A feedback methodology was developed to calibrate bridge crossing screen-line counts. This methodology uses feedback between the Activity Generator and itself.
- 3) The last feedback loop is to correct the mode choice from that given by the Activity Generator. In Fig. 1, this is represented by the feed-forward arrow from the Activity Generator to the Route Planner and the feed back arrow in the reverse direction.

These three feedback loops are explained in detail in Volume Three (*Feedback Loops*) of the Portland Study Report series of documents.

In TRANSIMS, models are developed from a series of feedback loops between the TRANSIMS modules that changes the behavior of selected individuals in the synthetic population. The order in which the modules are called in one of these loops constitutes a meta-method. Each meta-method has an associated method that controls the changes made to the selected individuals in the synthetic population. Neither the meta-methods nor the methods are part of the core TRANSIMS technology. Some of the meta-methods and methods used in the study are depicted in Fig. 2. These methods are described in detail in the Portland Study Report series.

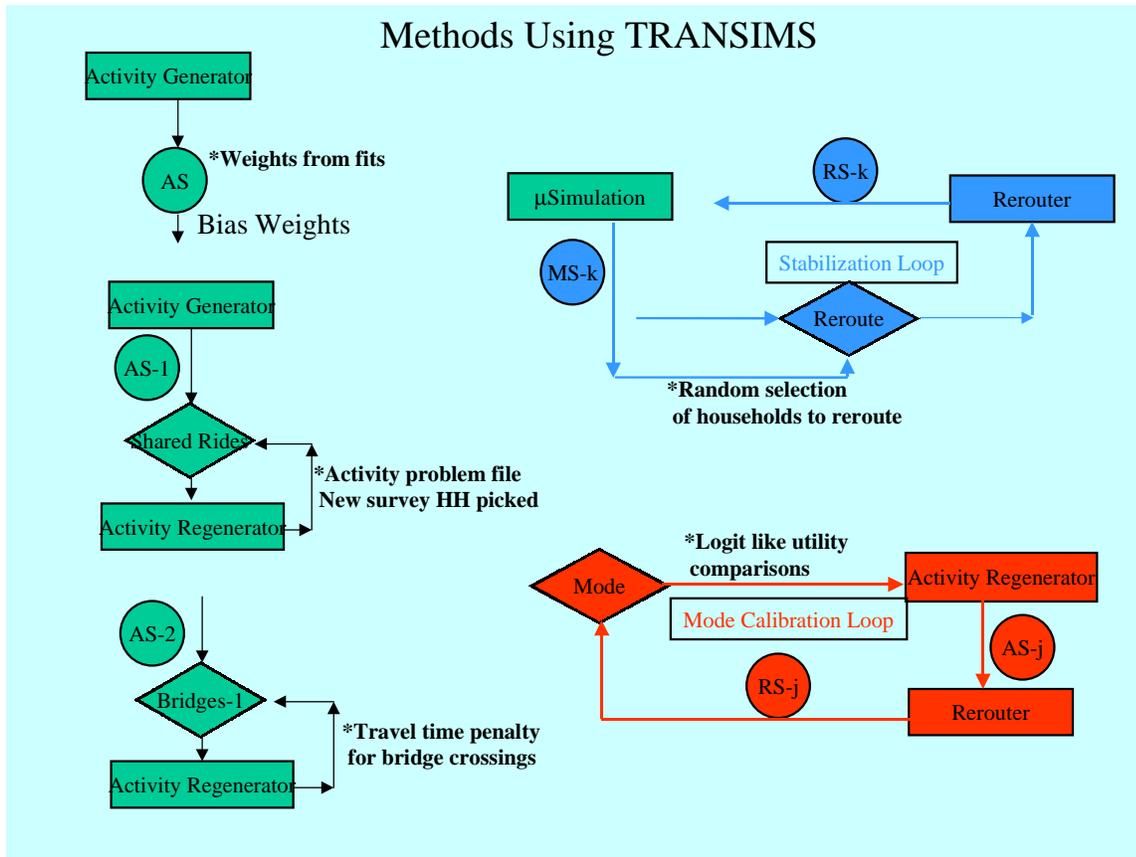


Fig. 2. Data from the TRANSIMS modules are fed back in systematic ways by meta-methods. Each of the meta-methods contains a method for handling the data that is fed back. This picture shows some of the meta-methods and the associated methods (denoted with an asterisk (*)) used in the Study. Both the meta-methods and the methods are external to the core TRANSIMS technologies.

5. PORTLAND STUDY FLOW OF RUNS

Each model using the TRANSIMS technology is a series of feedback loops and the associated meta-methods and methods. The model for the Portland Study is outlined in Fig. 3.

- The circles represent data sets generated by the various modules:
 - AS-* are activity sets,
 - RS-* are route sets, and
 - MS-* are output from the microsimulation.
- The symbols (PS-F, AS-F, RS-F, MS-F, MS-E and EM-F) are the final data sets for the population, activities, routes, microsimulation, microsimulation with output for emissions calculations, and the emissions estimates, respectively.
- The diamonds signify the selection of specific individuals or households for changes in their activities or their routes.
- The rectangles are the various TRANSIMS modules.

The diagram shows three types of modeling.

- The first section represents the initial population and activity generation and a series of feedback loops into the Activity Regenerator to refine the initial activity set.
- The second section shows the feedback between the Route Planner (rerouter) and the Traffic Microsimulator for traffic stabilization.
- The third section gives the final mode choice feedback loop between the rerouter and the Activity Regenerator.

Activity sets AS-2 to AS-7 are refinements of the original activity set AS-1. They are each created by selecting specific households from the preceding activity data set and refining their activities. The following is a summary of these activity data sets.

- AS-1—This is the original activity set. The input data to generate this data set is presented in Volume Two (*Study Setup: Parameters and Input Data*), Chapter Three (*Activity Generator*) of this series of reports.
- AS-2—Households with infeasible shared rides were changed to produce AS-2. Activity patterns for households are generated by using the activities from a survey household as the activities of the synthetic household. Shared rides may become infeasible when, for example, the activities of a survey household with two adults and two children are used as a template for a household with one adult and three children. Suppose that in the two-adult household, the second adult spends his/her day taking the children from place to place. In the single adult household, the person to drive these shared rides is not available. Therefore, the shared rides become infeasible. Feasible activity sets are generated in these cases by resampling the survey households.

The Activity Generator picks the closest school of the appropriate type for children with school activities. If in AS-1, the child walks a long distance to school, the walk mode for that trip is changed to school bus in AS-2. Both the shared ride and school bus changes are discussed in Volume Three (*Feedback Loops*), Chapter One (*Shared Rides / School Buses*) of the Portland Study series of reports.

- AS-3—The most important screen lines in this study are those that cross the bridges over the rivers. In particular, vehicle trips that cross the Columbia River are critical because there are only two routes across the river. This data set was produced by iterating a penalty function for vehicle tours that cross the rivers. The result of this iteration was fine tuned to produce activity set AS-7. A complete description of this procedure is in Volume Three (*Feedback Loops*), Chapter Two (*Calibration of River Crossing Screen Lines*) of the Portland Study series of reports.
- AS-4—This activity set was generated by changing the mode from transit to drive where the distance from transit at the destination end was greater than some predetermined distance. A complete description of this procedure is in Volume Three (*Feedback Loops*), Chapter Three (*Experimental Mode Choice and Long Walks*) of the Portland Study series of reports.
- AS-5—Transit mode was changed to park-and-ride for those transit trips where the origin was far from a transit stop while the destination was close to a transit stop. A complete description of this procedure is in Volume Three (*Feedback Loops*), Chapter Three (*Experimental Mode Choice and Long Walks*) of the Portland Study series of reports.
- AS-6—If a long walk was requested between activities, the location of the second activity was moved closer to the first. A complete description of this procedure is in Volume Three (*Feedback Loops*), Chapter Three (*Experimental Mode Choice and Long Walks*) of the Portland Study series of reports.
- AS-7—This data set is a refinement of the bridge-crossing procedure. A complete description of the bridge-crossing procedure is in Volume Three (*Feedback Loops*), Chapter Two (*Calibration of River Crossing Screen Lines*) of the Portland Study series of reports.

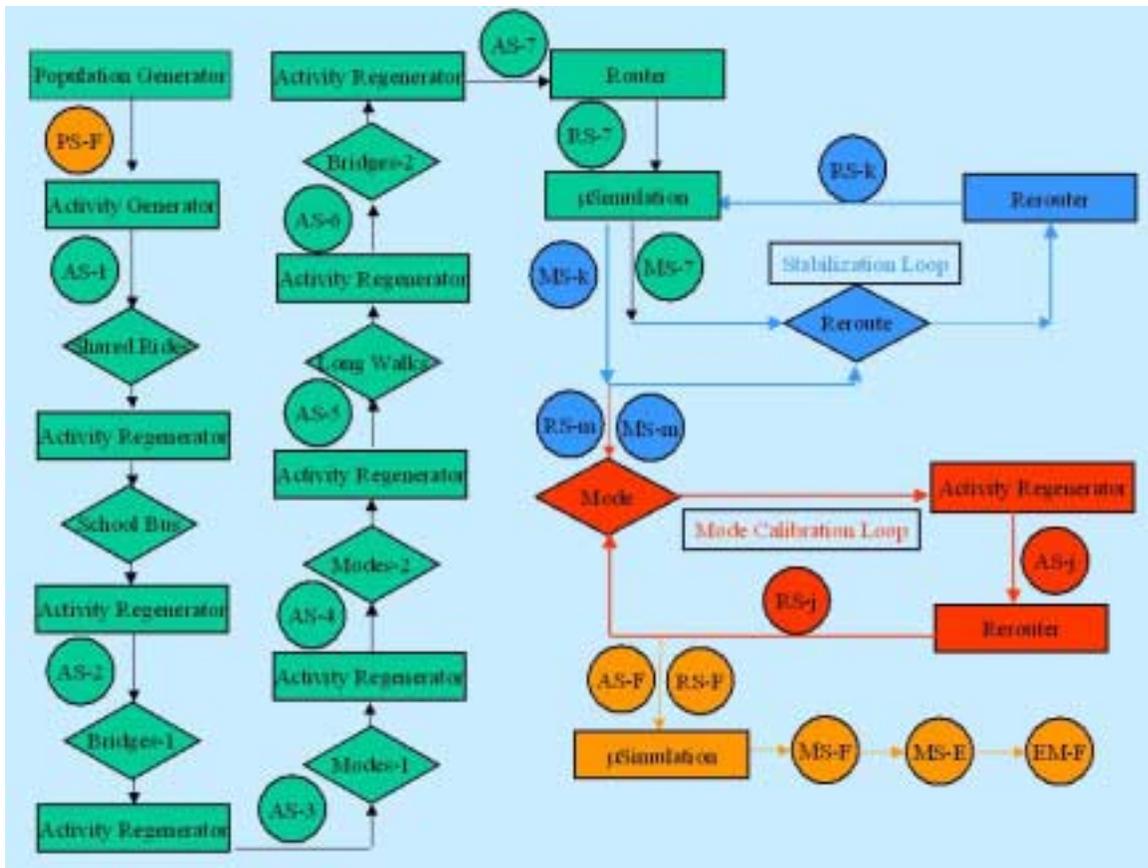


Fig. 3. Each model using the TRANSIMS technology is a series of feedback loops. The model for the Portland Study is outlined in this graph. The circles represent data sets generated by the various modules. AS-* are activity sets, RS-* are route sets, and MS-* are output from the microsimulation.

The stabilization loop produces route sets, RS-7 to RS-19, and the corresponding microsimulation data sets, MS-7 to MS-19. The stabilization procedure is described in Volume Three (*Feedback Loops*), Chapter Four (*Stabilization*) of the Portland Study Report series.

The red loop is critical. Iterations between the Route Planner and Activity Generator (activity regenerator) using a set of utility functions produces a calibrated mode split. The procedure for this calibration is given in Volume Three (*Feedback Loops*), Chapter Five (*Mode Choice*) of the Portland Study Report series.

In an actual study, after the mode selection has been made and the trips routed by the correct mode, the final activity set (AS-F) and route set (RS-F) are produced. These are microsimulated to give MS-F. A last microsimulation run is completed where environmental data is collected, MS-E, and the emissions are estimated in EM-F.

The general results of these simulations are found in the Portland Study Report series Volumes Four (*General Results (not validation)*) and Five (*Environmental Results*). A comparison to Portland data is given in Volume Six (*Comparisons to Portland Data*).

6. THE COMPUTING ENVIRONMENT

TRANSIMS modules are designed to use the power of parallel computation to decrease the time required for module execution and to leverage modern computing systems.

Two types of parallel computing systems were used for the Portland Study:

- 1) Linux cluster using Intel processors from VA Linux
- 2) Shared memory multi-processor system from Sun Microsystems

6.1 Linux Cluster

Hardware configuration:

- 64 dual-processor nodes with Intel Pentium III 500 MHz processors
- 1 gigabyte of memory per node
- 100 megabit/sec Ethernet connection between the nodes
- RedHat Linux 6.2 operating system (Linux 2.2.16 kernel)
- 100 gigabyte disk accessible from all nodes via NFS

Most of the runs for the Portland Study were made on the Linux cluster because of the number of processors available even though instabilities in this research cluster caused many problems during the study.

6.2 Shared Memory Multi-Processor System

Hardware configuration:

- Sun Microsystems Enterprise 4000 with 14 Ultra Sparc II processors, 248 MHz
- 6 gigabytes of memory
- 300 gigabytes of disk
- Solaris 7 operating system

The Sun Enterprise was used for code development, data analysis, network construction, and computations for the Portland Study that did not require more than 14 processors. This hardware configuration was very stable and delivered consistent performance.

Part 2

Parameters and Input Data

Chapter 1

Network

1. INTRODUCTION

The TRANSIMS network tables were created from data supplied by Portland Metro. The data, provided in ESRI Shapefiles, were reprojected from the Oregon State Plane coordinate system (Zone 5076) with measurements in feet to Universal Transverse Mercator (UTM) coordinates (Zone 10) with SI units. Both projections utilized the NAD83 datum. ESRI's ARC/Info 7.1.1 and ArcView 3.2 were extensively utilized to manage the geographic data. After the tables were completed, they were checked with the *NetworkValidator* utility (see the general documentation for more information). The tables utilized in the case study ranged in size from 19 kilobytes (KB) to 35.6 megabytes (MB), depending on the numbers of records and attributes each table contained. Table 1 provides a list of table sizes and number of records.

Table 1. Size and number of records contained in each TRANSIMS network table.

Table	Number of Records	Size (MB)
Activity Location	243,423	35.6
Detector	10,830	0.5
Lane Connectivity	507,700	12.7
Link	124,904	16.8
Node	100,539	3.5
Parking	121,504	8.5
Phasing Plan	33,871	1.1
Pocket Lane	4,379	0.16
Process Link	526,094	24.4
Signal Coordinator	2,086	0.02
Signalized Node	2,081	0.08
Timing Plan	1,716	0.05
Transit Stop	9,827	0.56
Unsignalized Node	237,343	4.7

to this threshold. Table 2 shows the minimum speeds used by each function class. Not included are 115 links that were assigned low speeds (4 or 5 m/s) to discourage travel on these streets in order to obtain more realistic travel patterns.

Table 2. Function class minimum speeds.

Function Class	Number of Links	Minimum Speed (m/s)
COLLECTOR	3231	11
FREEWAY	1604	36
LIGHTRAIL	58	11
LOCAL	104693	11
PRIARTER	5789	11
RAMP	1552	11
SECARTER	6839	11
WALKWAY	129	0
XPRESSWAY	894	11

- FUNCTCLASS
- VEHICLE – Autos, buses, and trucks were allowed on every link that was not specified as a bus-only or light-rail-only link in the EMME/2 network. Each link mode was further cross-checked with the transit routes to assure that the specified transit vehicle was permitted on the required links. Refer to Section 7 (*Transit Tables*) for more information. Pedestrians were allowed on all streets, except freeways, expressways, and ramps.

Any link without an equivalent in the EMME/2 network was assumed to be a local street. These links were assigned one lane in each direction with a speed limit of 25 mph (11 m/ps), a capacity of 500 vehicles per lane, and the function class LOCAL.

The remaining attributes were manually added or generated by Metro or LANL:

- NODEA/B
- LEFTPCTSA/B and RGHTPCKTSA/B
- TOWAYTURN
- LENGTH – Calculated by ArcView and/or ARC/Info.
- GRADE – Based on the elevations of the link end nodes (Node A and Node B). It was calculated by taking the difference in elevations from Node B to Node A and dividing it by the link length. Refer to Section 3 (*Node Table*) for more information on node elevations.
- SETBACKA/B – Set to zero for the case study.
- FREESPDA/B – On most links, the vehicles were allowed to travel at speeds up to 1.25 times the link’s speed limit.
- THRU A/B – Default “through” link was determined by the *PrepareNetwork* program and automatically added to the link table.

- COLOR – No longer utilized by the microsimulation so was given the default value of zero.
- NOTES – Given the default entry NONE.

The completed link table, which contains approximately 124,900 records, is 16.8 MB.

2.2 Problems

The link network generally contained two error types: human mistakes and inaccuracies built into the data based on its source. Errors caused by human interaction included:

- Reversed lane connectivity
- Inconsistent link attributes, such as the speed limit and number of lanes, for continuous sections of roadways encompassing multiple links

The main source of error is due to the fact that the network is based on the TIGER street data. The TIGER data, which historically have been digitized from the U.S. Geological Survey (USGS) topographic maps for the Census Bureau, contain many topological errors and are currently not designed for this type of application.

- Several ramps, cul-de-sacs, and other road segments began and ended at the same node. Since the microsimulation acquires a link's geographic location from its end nodes' coordinates, a link that begins and ends at the same place appears to be only a point. Many of the local street cul-de-sacs were eliminated since this would not greatly alter the microsimulation results. More important features, such as ramps, were divided into multiple links by inserting new nodes as shown in Fig. 2

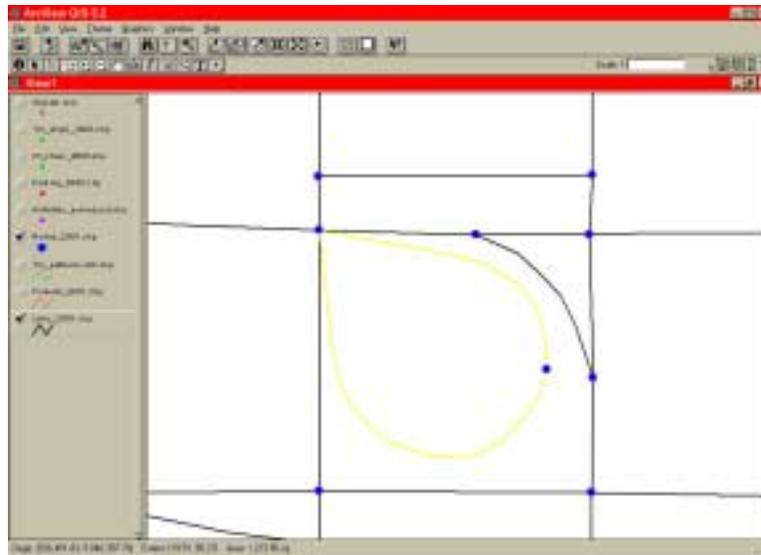


Fig. 2. Ramp that was both divided into two pieces and had a node split into two to separate the overpass and the underpass links.

- Because the TIGER network is only a two-dimensional data set, overpasses and underpasses are connected together by nodes, which were inserted by ARC/Info whenever two road segments crossed. Also, a ramp (Fig. 2), comprised of multiple, continuous links, is sometimes connected to both an underpass and an overpass by the same node. If not corrected, simulated vehicles could legally drive off an overpass onto an underpass and avoid any ramps. To resolve these situations, the node in question was divided into two or more nodes as needed; each node exists at a different elevation. Refer to Section 3 (*Node Table*) for more information.
- Multiple links that have matching beginning and ending nodes, such as a side street shown in Fig. 3 that branches off a main road and then rejoins it, appear in the microsimulation to be duplicated copies of the same link. New nodes were inserted into the secondary links to divide them into new links with different end nodes.

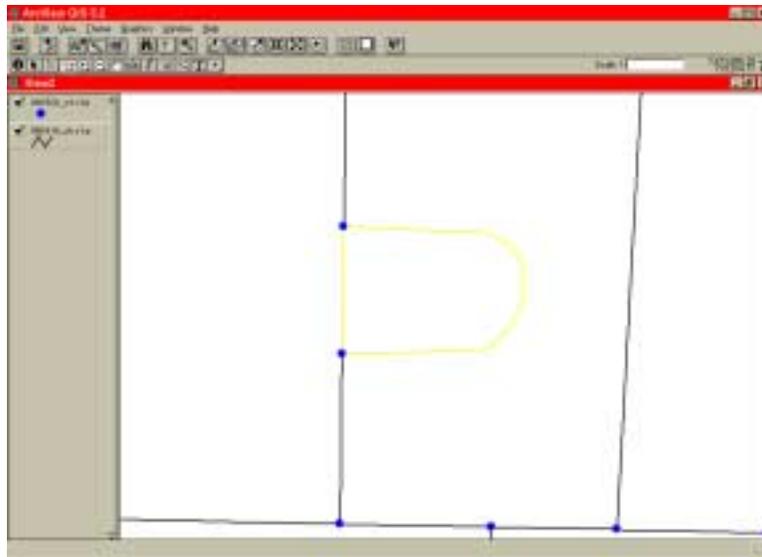


Fig. 3. Two links sharing the same end nodes.

- At some nodes, links appeared to be connected to one another but were in fact not attached. Instead, each of the links ended at a different node; the nodes were located in close proximity to each other. The links, such as those in Fig. 4, had to be manually joined together, or simulated vehicles could not travel across the disjointed links.

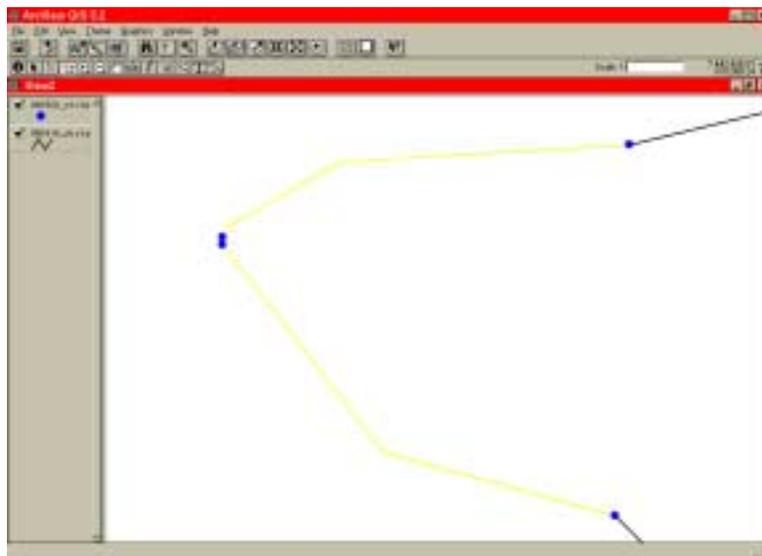


Fig. 4. Two links' disjointed links that should be connected.

- In a few instances, street segments were composed of overlapping links. Instead of having a series of links that connected end to end, sections of the links were repeated by other links. Since this could cause serious problems when routing trips, the overlapping links were removed.

- Links with a very short length caused various traffic problems. In some cases, vehicles did not have enough room to move from one lane to another when needed, such as when a vehicle has to move into a turn lane. This resulted in traffic jams from the vehicle becoming stationary or the vehicle becoming lost. Because of editing time restraints, all links less than five meters were removed from the network by merging them with longer links. Links that ranged in length from five meters to less than 15 meters were assigned a link length of fifteen meters. The topology of these links was unchanged. A further length modification involved extending the freeway and expressway links. The length of any freeway or expressway link that had a length less than the sum of the link's length and speed limit was increased to that value.
- Intersections that were comprised of multiple links converging at multiple nodes needed to be merged into one node. For example, a divided expressway, as seen in Fig. 5, is often represented with two one-way links. At areas where another road intersects the expressway, a small link is needed to cross from one expressway section to the other, thereby requiring the intersection to be comprised of multiple nodes. For traffic signals and signs to operate properly, these multiple node intersections must be condensed down to join at only one node.

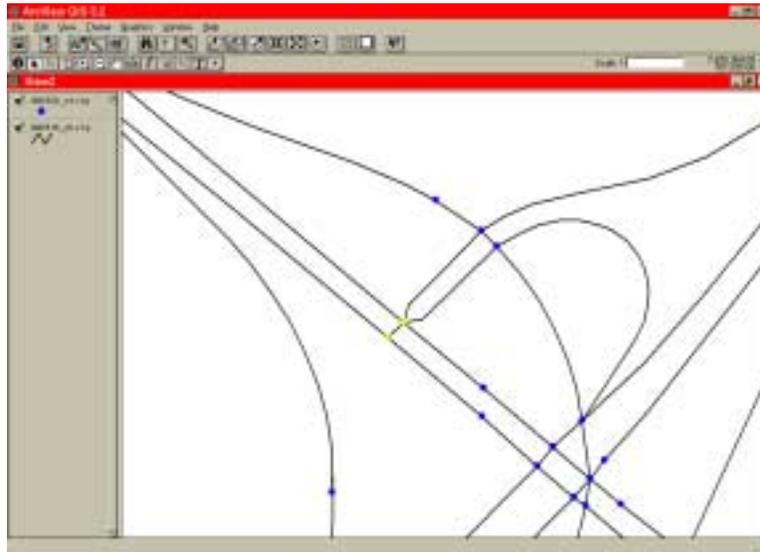


Fig. 5. An intersection with a signal that is spread over multiple nodes.

- Alleys, contained in the TIGER data, were eventually removed from the network since the links were short and would have had a minimal impact on traffic flows.

3. NODE TABLE

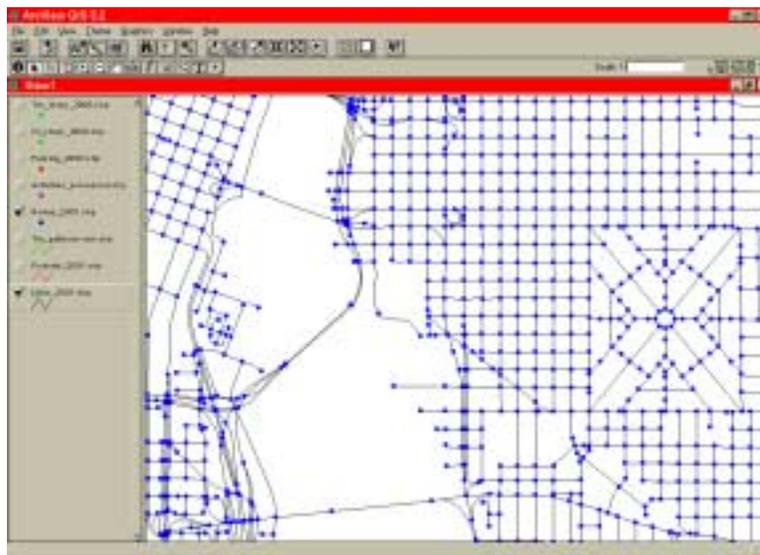


Fig. 6. Nodes in the Portland network.

3.1 Data

Along with the link data, the node data, shown in Fig. 6, are also part of the TIGER road data set. The two-dimensional data set was supplemented by the addition of the elevations for each node. The TIGER node data contained the required TRANSIMS data.

- ID
- EASTING – Inserted using the ArcView commands [Shape].GetX in the Calculate function.
- NORTHING – Inserted using the ArcView command [Shape].GetY in the Calculate function.
- ELEVATION – Initial node elevations' values were determined from the region's soil surface elevation grid. Surface road elevations were assumed to be approximately equal to the soil surface elevations. The elevations for below- and above-grade structures, including ramps, overpasses, bridges, and tunnels, were estimated based on the soil surface elevations. At locations where links cross but in reality do not actually intersect (i.e., overpasses and underpasses), each node was split into multiple nodes existing at different elevations. Aerial photographs and fields checks were utilized to confirm which links crossed above or below other links that these intersections. The node elevations of these links were also checked to verify that the values provided sufficient vehicle clearance where appropriate. Bridge elevations for the Willamette and Columbia Rivers were calculated from available maximum height values that were added to the water surface elevations.
- NOTES – Given the default entry NONE.

The completed node table, approximately 3.5 MB in size, contains nearly 100,500 records.

3.2 Problems

As with the TIGER link network, the node problems were related mainly to conflicts between the TIGER and TRANSIMS formats. In areas where overpasses and underpasses cross, the two-dimensional TIGER network places a node, forcing the links to intersect. During the initial splitting process, several nodes, which needed to be separated, were missed. This resulted in the nodes having to be divided at a later date during testing, which in turn affected the microsimulation results.

4. POCKET LANE TABLE

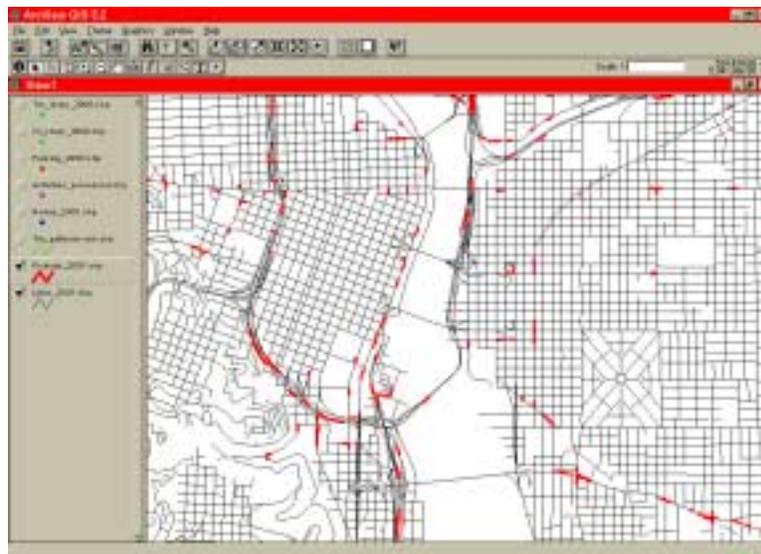


Fig. 7. Pocket lanes in the Portland network.

4.1 Data

The pocket lanes (Fig. 7) were created by Metro based on a series of four-foot resolution aerial photographs covering the entire Portland metropolitan area. Metro created a tool in ArcView that allowed the user to draw and capture the pocket lanes as they appeared in the aerial photos. The tool also prompted the user to input attribute information—which node the pocket lane led towards, side of street the pocket is located, type of pocket lane, etc. This information was output into an ArcView shapefile, which in turn was transformed into the pocket lane table. Data output by Metro’s tool for direct use in the pocket lane table included:

- ID
- NODE

- LINK
- OFFSET – Measured using an ArcView tool.
- STYLE – All pocket lane styles were used.
- LENGTH – Initially calculated by ArcView.

Other fields added later were:

- LANE – The lane number was determined from a combination of the number of permanent lanes and pocket lanes on one side of a link. The number of right and left pocket lanes for each type on a side of a link was summed. Since one turn lane, merge lane, and pull-out can each have the same lane number, assuming they do not overlap, the largest sum value for each of the right and left pockets was saved in the links table as the number of pocket lanes in a particular direction (LEFTPCKTA, RGHTPCKTB, etc.). The lane numbers in the pocket lane table are based on the number of lanes and the permanent lanes. Because lanes are numbered from the middle to the outside of a link, left turns are numbered first (i.e., lane 1, lane 2, etc.). Permanent lanes are numbered next (lane 3, lane 4, etc.), and right pocket lanes are numbered last (lane 4, lane 5, etc).
- NOTES – Given the default entry NONE.

The completed pocket lane table, which contained approximately 4400 records, was 158 KB.

4.2 Problems

Because of the way the TIGER road network fragmented the streets into link segments, several problems emerged while creating the pocket lanes.

- Pocket lanes sometimes, in reality, crossed several links. Because TRANSIMS pocket lanes can exist only as a portion of one link, the pocket lane had to be redrawn to accommodate the network topology. Further time for link topology improvements would have resulted in more accurate pocket lane lengths.
- Several features, such as large parking lots, often utilize turn pockets for entering and exiting. However, if no street leading to the parking lot exists in the street network, these turn pockets should not be included in pocket lane data set. Turn pockets of this type that were included in the pocket lane table were removed or changed to pullout lanes.
- Intersections that contained multiple offset links, such as divided highways, sometimes had the pocket lane assigned to the wrong link. For example, the small link between the two divided highway links was forgotten, so the turn/merge lane was placed in the wrong location.

Other problems included:

- The incorrect NODE was listed in the table, so the microsimulation placed the pocket lane at the wrong end of the link.

- Pocket lanes were not updated when the link network data set was corrected. This sometimes resulted in pocket lanes no longer being connected to intersections with turns because a link was inserted between the pocket lanes and the intersection.
- Merge lanes were added to all freeways connected to ramp links in order to obtain more realistic traffic patterns.

5. ACTIVITY LOCATION TABLE

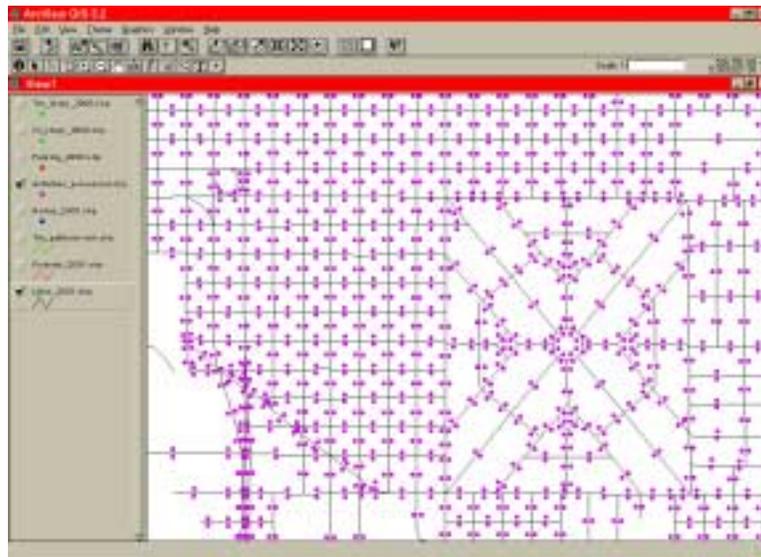


Fig. 8. Activity locations in the Portland network.

The activity locations, shown in Fig. 8, utilized in the case study are generic points representing housing and work sites. Two activities are placed on each link; each activity is located ten meters to one side of the link mid-point. Generally, all links have activity locations except bridges, ramps, and freeways. Extra boundary activities, used for routing, were placed in specific traffic zones and on network edges where no activity locations existed, such as on a freeway.

Each activity location has two types of data associated with it: required and optional. The mandatory data set includes information that provides the position of the activity locations.

- ID
- NODE
- LINK
- OFFSET – Placed in the center of the link so the offset was half the link length.
- LAYER – The WALK layer was utilized as the default mode.
- EASTING – Inserted using the ArcView commands [Shape].GetX in the Calculate function.

- **NORTHING** – Inserted using the ArcView command [Shape].GetY in the Calculate function.
- **NOTES** – Given the default entry NONE.

The accessory data, utilized to route travelers, are optional. The type of data included, if any, is up to the user to define. For the Study, one set of activity data was created from various types of geographic data. This data set was then processed to create the attractors employed when routing travelers throughout the network. The original unprocessed activity location attributes are:

- **Traffic Analysis Zones (TAZ)** – Provided by Metro. The TAZ values were added to the activity file using *identity*, a point-in-polygon command in Arc/INFO. The activity location is the point, and the TAZ data are the polygons.
- **Census Tract and Block Group Numbers** – Added using the same procedure as in the inclusion of the TAZ values.
- **Households** – The number of households on a link was provided by Metro employing the following procedure:
 - 1) Create a database of single- and multi-family parcels from digital tax lot data.
 - 2) Calculate the percentage of total parcel area that one parcel occupies inside a TAZ.
 - 3) Allocate households to each parcel based on parcel size by multiplying the total number of households in a TAZ by the parcel percentage calculated in Step 2.
 - 4) Distribute parcels to the street network. The links, excluding freeways and ramps, were segmented into smaller units based on each link vertex. Parcel centroids were matched to the nearest segment midpoint. The link segmentation improved the occurrence of the parcels attaching to the appropriate link.
 - 5) Sum the number of households on a link from the link segments.
 - 6) Since there are two activities per link, the number of households on a link was equally divided between both activities.
- **Schools** – Provided by Metro. Each public and private elementary and second school was assigned to the nearest activity location. The activity location table shows what type of school is assigned to each activity: 0 = none, 1 = elementary, 2 = middle/junior high school, and 3 = high school.
- **College and University Enrollments** – Allocated to the activity location nearest to the schools' geographic positions. Some schools were spread over multiple campuses so each area was given a percentage of the total enrollment of the school.
- **Number of Employees** – To determine attractors for work locations, Metro provided the locations of companies operating in the case study area. Each company was assigned to the nearest activity location. The businesses were characterized by their Standard Industry Classification (SIC) code. Then, the total employees for each SIC code category (Service, Retail, Government, College, and Other) at an activity location was summed. These SIC

category counts, as well as the total number of workers employed at that location, were included in the activity location table.

- Park Area – The total area surrounding an activity location that is designated park or recreation land was supplied by Metro.
- Number of Households Per Acre – Calculated by first summing the total number of households in each TAZ. Each sum was divided by its respective TAZ’s total area (in acres). Finally, the activities were assigned a value based on the TAZ in which it was located.
- Distance to Nearest Transit Stop – The Euclidean distance between an activity (X_A, Y_A) and the nearest transit stop (X_T, Y_T) was calculated using the following formula:

$$Dist = \sqrt{(X_A - X_T)^2 + (Y_A - Y_T)^2}$$

- Trip Origin/Trip Destination – Because some of the boundary activity locations have been placed on the edge of the network on one-way links, such as freeways, routing problems resulted in some travelers not being able to complete their trips. These Boolean columns were utilized to show whether each activity could be used as a trip origin or destination location to avoid this problem.
- River Zone – Arbitrary zone number assigned for analysis purposes.
- Urban Type – Arbitrary zone number assigned for analysis purposes.
- Parking Zone – Arbitrary zone number assigned for analysis purposes.

The initial activity location attributes were processed with an algorithm to generate a new set of attractors. The number of households, locations of schools, college and university enrollment, and the number of employee types were replaced with: Home, Shop, Visit, Social, Other, Serve Passenger, and College. The land use and employment variables used to create these attractors are described in Volume Two (*Study Setup: Parameters and Input Data*), Chapter Three (*Activity Generator*) of the Portland Study series of reports.

The other original attributes were included in the final activity location table without being altered. The final table, which contained approximately 243,400 records, is 35.6 MB.

6. PARKING

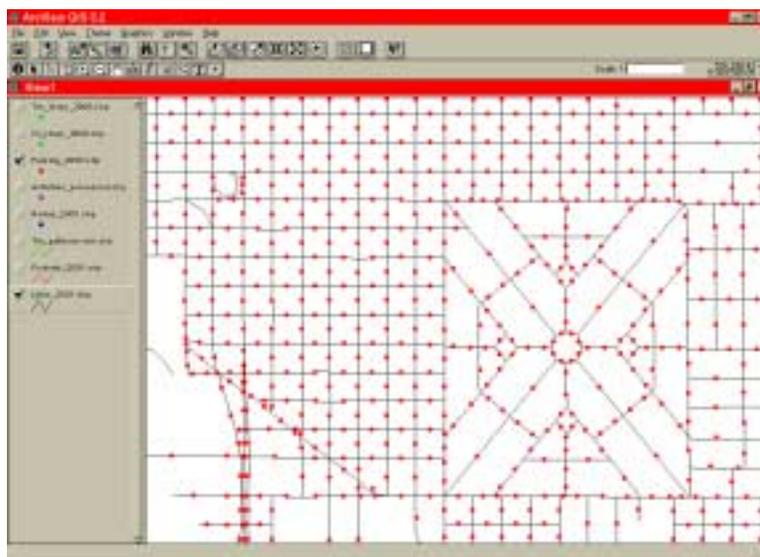


Fig. 9. Generic parking locations in the Portland network.

Parking locations, lots, and network boundary places, were created for the microsimulation. Generalized parking lots (Fig. 9) were placed on links not designated as ramps, bridges, freeways, or walkways. Special parking lots were created for some bus-only and light-rail-only links so transit vehicles would have a place to begin and end transit routes. Network boundary parking locations were added to specific boundary zone areas needed for modeling purposes and at network edges where parking did not exist, such as on freeways. Finally, generic park and ride lots were added to links with actual park and ride lots, the locations of which were provided by Metro. The parking table contains the following data:

- ID
- NODE
- LINK
- OFFSET – Placed in the center of the link so the offset was half the link length.
- STYLE – Lots, park and ride, and network boundary parking were used.
- CAPACITY – All lots were assigned unlimited parking capacities.
- GENERIC – Used in order to simplify the microsimulation.
- VEHICLE – Modes allowed were autos, buses, light rail transit, and trucks.
- STARTTIME/ENDTIME – This feature is not currently being utilized by microsimulation, so there are no enforced parking restrictions.

- NOTES – Given the default entry NONE, except for boundary lots and those lots used only by transit vehicles.

The parking table, which contains approximately 121,500 lots, is 8.5 MB.

7. TRANSIT TABLES

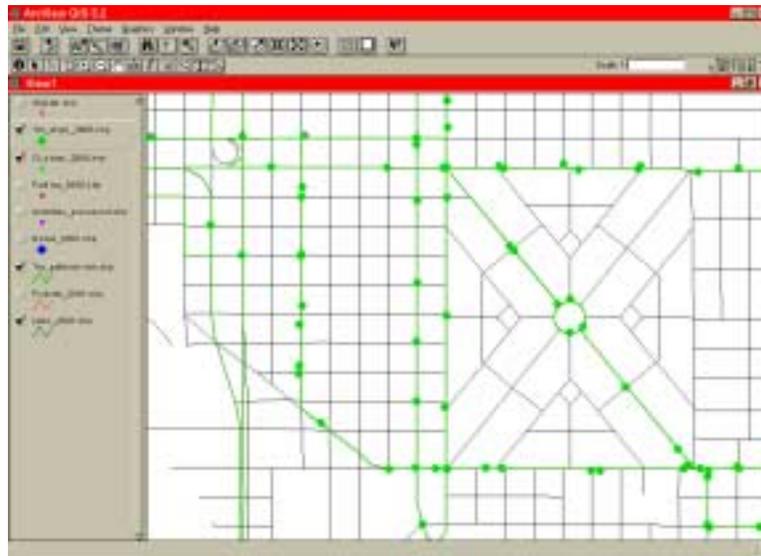


Fig. 10. Transit stops and routes in the Portland network.

Metro collected the transit data from two agencies within the Portland metropolitan area: Tri-Met, which primarily serves the Oregon counties, and C-Tran, which provides service mainly to Clark County in Washington.

7.1 Transit Stop Table

7.1.1 Data

The transit stops for most of Portland were provided to Metro by Tri-Met in ArcView shapefiles, shown in Fig. 10. Initially, the Tri-Met stops contained multiple records for the same stop if it was utilized by more than one transit route. A custom program, written by Metro, created the transit routes (see Section 7.2 (*Schedule File*)) and output the transit stop information, with one record per stop, in the proper TRANSIMS format. The transit stop data provided to Metro by C-Tran was in text format. Transit stop locations were manually geocoded onto the link network. Metro provided a condensed version of all transit stops, containing only one record per stop as required by TRANSIMS formatting.

The transit stop table was created with the following fields:

- ID

- NAME
- NODE
- LINK
- OFFSET –The custom routing tool that created the master transit stop list also calculated the link, node, and offset information and saved the information.
- VEHICLE – Only buses and light rail transit were modeled.
- STYLE – Only stops used; no stations were included.
- CAPACITY - Set to zero for an unlimited number of vehicles allowed at a transit stop.
- NOTES – Given the default entry NONE.

The transit stop table, which contains approximately 9800 stops, is 563 KB.

7.1.2 Problems

Because of errors in the link network, several of the transit stops were coded on the wrong side of the street. This caused problems later on when trying to build the Transit Route file and the Transit Driver plans. To correct the problem, the stop was moved to the other side of the street by changing the `to` node in the transit stop table.

7.2 Schedule File

7.2.1 Data

The schedule information for the Oregon buses and light rail vehicles was provided to Metro by Tri-Met. The file contained the time each transit vehicle stopped at every transit stop along all routes for an entire day on a typical weekday. C-Tran supplied Metro with a copy of their printed bus schedule and a list of all stops in a bus route, including those stops not listed on the schedule. Metro provided LANL with the complete Tri-Met itinerary and an electronic version of the C-Tran printed schedule, as well as the list of stops for each route.

To complete the C-Tran itinerary, a custom C++ program was written that calculated the stop times for all stops in each route based on the printed time table, the complete list of stops in each route, and links composing each bus route. The time to travel between each stop in the route was calculated based on the length and speed limit of each link in the route. This time was then added to the known stop times in order to estimate the time of arrival at the other stops. A final itinerary for the C-Tran buses was output from the program. The itineraries for the Tri-Met and C-Tran transit vehicles were combined and sorted (in the order shown below) to create the transit schedule file.

- Route ID
- Time

- Stop ID

The final table, which contained approximately 495,000 records, was 7.4 MB.

7.2.2 Problems

When the links and nodes composing each transit route were compared with the order of stops visited in each route, several of the stops were discovered to be located outside of the route path or placed on the wrong side of the street. The stops were either moved onto the path of the transit route by moving it to a new link and/or node, or the stop was removed from the transit schedule.

7.3 Transit Route Table

7.3.1 Data

The transit routes for the Tri-Met and C-Tran data, as seen in Fig. 10, were created by a custom ArcView Network Analyst tool written by Metro. The tool determined the shortest path between transit stops to construct the entire transit path on the link network. In the instances when the shortest path algorithm did not properly replicate the transit route, dummy stops could be added manually to force the path to use the correct network links. Metro provided the transit route data in shapefiles and database tables so the route could be represented by a list of links and to nodes in order. A custom C++ program was written to check the routes to ensure that the connectivity was correct. This included verifying that the links and nodes were connected to each other in the proper order, and that each link had at least one lane for the transit vehicle to drive on. The program also output the transit route table in the proper format.

- Route ID – Assigned by Metro when assembling routes.
- Number of Stops in Route – Summed by C++ program.
- Transit Type – Bus or Light Rail.
- Stop ID
- Link ID
- Node ID
- Transit Zones –Determined from the Tri-Met and C-Tran transit fare zones. A separate table, the Transit Zone table, contains the costs of traveling between each zone.

The table, containing 626 routes, is 623 KB in size.

7.3.2 Problems

Because of link network repairs, the transit routes sometimes had connectivity problems. The connectivity of the route links and nodes were checked frequently to ensure that the route paths were correct.

7.4 Transit Driver Plans

7.4.1 Data

The transit driver plans were assembled by a custom C++ program. The program combines the transit route node paths, transit schedule starting and ending times, parking data for route starting and ending locations, and the link data to include path information to and from the starting and ending parking lots into the format required by the microsimulation.

- Traveler ID – The traveler IDs for the transit drivers were sequentially numbered, beginning with ten million. This number was chosen so that the transit drivers would not have the same traveler IDs as the synthetic population.
- User Field – Given the default value 0 since this is not used by the microsimulation.
- Trip ID – Assigned the default value 1 since each transit driver plan contained only one trip.
- Leg ID – Given the default value 1 since each trip only had one leg.
- Activation Time – Calculated by subtracting 45 seconds from the time the transit vehicle leaves the first stop.
- Start Accessory ID – All transit routes began at a parking lot. The custom C++ program located the parking lot nearest to the first transit stop and assigned the parking lot to be the starting accessory.
- Start Accessory Type – All start accessories were parking lots, so the default value 2, which represents parking lots, was used.
- End Accessory ID – All transit routes end at a parking lot. The custom C++ program located the parking lot nearest to the last transit stop and assigned the parking lot to be the end accessory.
- End Accessory Type – All ending accessories were parking lots, so the default value 2, which represents parking lots, was used.
- Duration – Determined by subtracting the Activation Time from the Stop Time.
- Stop Time – Calculated by adding 45 seconds to the time the transit vehicle arrives at the last stop in the route.
- Max Time Flag – Boolean set to TRUE.
- Cost – Given the default value of zero.
- CGF – Same as the Duration, so that value was inserted.
- Driver Flag – Boolean set to TRUE since the plan is for the transit driver.
- Mode – Set to 1, since this is the value used for the transit mode of travel.

- Number of Tokens – The number of values in the mode-dependent data for a transit driver, including a count for the token itself, the Schedule Pairs digit, Vehicle ID, Route ID, number of nodes in the route, and the number of Stop IDs and Departure Times in the Schedule Pairs.
- Schedule Pairs – No schedule information was used to route the transit vehicles, so this was set to zero.
- Vehicle ID – Set to the same number as the Traveler ID in order to avoid conflicts with the auto vehicle ids.
- Route ID
- List of Node IDs – The nodes, in order, were gotten from the route data. If the beginning and ending parking lots were not on the same node as the beginning and ending transit stops, those nodes were added to the list of nodes.
- List of Schedule Pairs – Not used in the microsimulation.

The Transit Driver Plan file, which contains approximately 9400 plans, is 9.1 MB.

7.4.2 Problems

Because the transit driver plans rely on data from so many data sources, any changes to the links or nodes in the transit path, transit schedule, or the parking could result in the transit driver plans being corrupted. With so many variables, it was sometimes difficult to track down problems. Also, if the beginning parking lot is located on the same link as the first transit stop, the first node in the transit route must be located past the parking lot in the route. When executing each transit driver plan, the vehicle leaves the starting parking location and heads towards the first node in the route. If the first node is before the parking lot, the vehicle will have to make a U-turn to continue onto the first transit stop. A similar problem occurs if the final node in the route is located after the ending parking lot in the route.

7.5 Transit Vehicle File

The vehicle plan file was created by a custom C++ program that input the transit driver plan file and output the vehicle plan file in the proper format.

- Household ID – Since no households were involved with the transit plans, the default value zero was used.
- Vehicle ID – Same as in the Transit Driver Plan file.
- ID of Starting Location – Beginning parking lot id.
- Vehicle Type – Light rail transit = 8, and buses = 5.

The file was 180 KB in size and contained approximately 9400 records.

8. PROCESS LINK TABLE

The extent of the case study network prohibited the inclusion of all possible process links in the microsimulation. Instead, a subset of process links was used. All activities were connected to the nearest parking location, with emphasis placed on matching activities and parking lots located on the same link. Process links were also used to join transit and activities that share a common link. If an activity had been placed on a link with no transit stops, no process links were created. The process link data included:

- ID
- FROMID
- FROMTYPE – Activity, Parking, or Transit Stop.
- TOID
- TOTYPE – Activity, Parking, or Transit Stop.
- DELAY – Set to 23 seconds.
- COST – No cost was incurred for traveling on the process links.
- NOTES – Given the default entry NONE.

The process link table, which contains approximately 526,000 records, is 24.4 MB.

9. LANE CONNECTIVITY TABLE

The lane connectivity table was generated by the *LaneConnectivity* program. However, there were approximately two dozen areas in the network with specialized connectivity that needed to be altered by hand. When analyzing the traveler plan files, it was obvious that certain links, particularly freeways and ramps that were expected to have a large number of travelers, in fact, had very few or no vehicles utilizing them. This was most often due to the fact that the *LaneConnectivity* program cannot account for turn barriers. For example, highways are sometimes connected to a secondary street by more than one ramp and the vehicles can turn only from one direction onto a ramp due to turn restrictions. The generated lane connectivity allowed traffic to turn both ways from the secondary street onto the ramps. So, the router-generated paths tended to put the traffic more on one ramp than the other for some beneficial reason, such as the ramp being shorter than the other. Manually changing the lane connectivity table to more realistic probability solved this problem. Also, several transit routes required U-turns because of the way the paths were constructed. The needed U-turns were also added.

- NODE
- INLINK

- INLANE
- OUTLINK
- OUTLANE
- NOTES – Given the default entry NONE.

The lane connectivity table, which has approximately 507,700 records, is 12.7 MB.

10. TRAFFIC CONTROLS

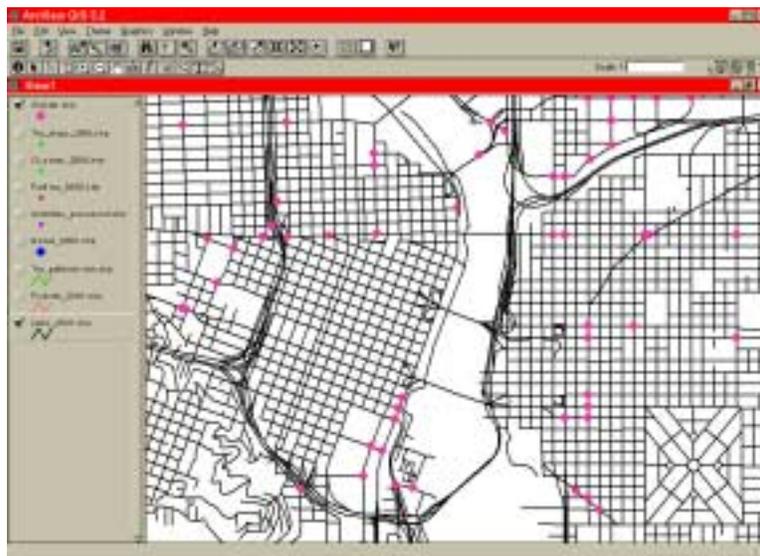


Fig. 11. Signalized nodes in the Portland network.

The traffic control tables (Unsignalized Node, Signalized Node (Fig. 11), Phasing Plan, Timing Plan, Signal Coordinator, and Detector tables) were generated automatically using the *CreateTrafficControls* program (see TRANSIMS Ver. 2.1 Volume Two (*Networks and Vehicles*), Section 1.6.3 (*CreateTrafficControls*) at <http://transims.tsasa.lanl.gov> for more information).

The default values for the following network configuration keys are as follows:

NET_ACTUATED_ALGORITHM_B_BETA	1 meters / second
NET_ACTUATED_ALGORITHM_B_DENSITY_CONST	0 meters
NET_ACTUATED_ALGORITHM_B_FLOW_CONST	0.1 / seconds
NET_DETECTOR_PRESENCE_SAMPLE_TIME	1 second

These values were obtained from the study of actuated signal behavior (see the general actuated signal behavior documentation). In order to take advantage of the concept that well-chosen green

times are necessary for coordinated signal operation, a signal-generation algorithm was developed where the minimum green length for a phase is based on a weighted sum of the number of incoming lanes participating in the phases, but it is never less than 25 seconds. (A permanent lane has a weight of 1, and a pocket lane has a weight of 1/3.) The total cycle length of at least 60 seconds is apportioned to the phases with constant yellow and red clearance intervals, and minimum green intervals based on the ratio of the number of incoming lanes in this phase to the total number of lanes. The green extension fraction is set to 60%, but the green extension is never permitted to be less than 12 seconds. Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Eleven (*MS-7*) of the Portland Study Report series lists the complete set of TRANSIMS configuration settings used to generate the signals.

Several modifications to the automatically generated traffic controls were made. Any yield signs that led from on-ramps onto freeways, as well as signals that were generated on freeway links, were removed. Intersections that encompassed multiple nodes in the network caused by offset links were combined into one node or the traffic controls were removed from most or all of the nodes to allow traffic to flow more realistically. Also, U-turns at signals were permitted where the connectivity allowed them.

Approximately 0.25% of the nodes were not accounted for in either the Signalized Node or Unsignalized Node tables. These nodes were generally one-way links (such as a freeway link) located on the edges of the network with traffic flowing in toward Portland. Because of this, none of the nodes had an “incoming” link, so they could not be assigned a record in the Unsignalized Node table. The link would also not require a signal with traffic flowing in only one direction.

The files created by the program varied in size from 19 KB (Signal Coordinator table) to 4.7 MB (Unsignalized Node table).

11. TRANSIMS TABLES NOT USED IN CASE STUDY

Several of the tables were not utilized in the case study:

- Speed Table
- Lane Use Table
- Barrier Table
- Turn Prohibition

Chapter 2

Population

1. OVERVIEW

TRANSIMS (Transportation Analysis and Simulation System) is an integrated system of travel forecasting models designed to give transportation planners accurate, complete information on traffic impacts, congestion, and pollution. The Population Synthesizer module constructs a regional population imitation with demographics closely matching the real population. Households are distributed spatially to approximate regional population distribution. The synthetic population's demographics form the basis for individual and household activities requiring travel, and their household locations determine some of the travel origins and destinations. This report outlines how we have constructed the synthetic population for our Portland, Oregon case study. It also briefly summarizes the characteristics of the data and how we verified that the data were correctly generated.

2. SOURCE DATA

Fig. 1 illustrates the data flow related to the TRANSIMS Population Synthesizer module. Input data for the module has been collected from a variety of sources: the TRANSIMS Population Synthesizer and Network user manuals provide detailed information on the organization of these input files—here we will only discuss where we obtained the data.

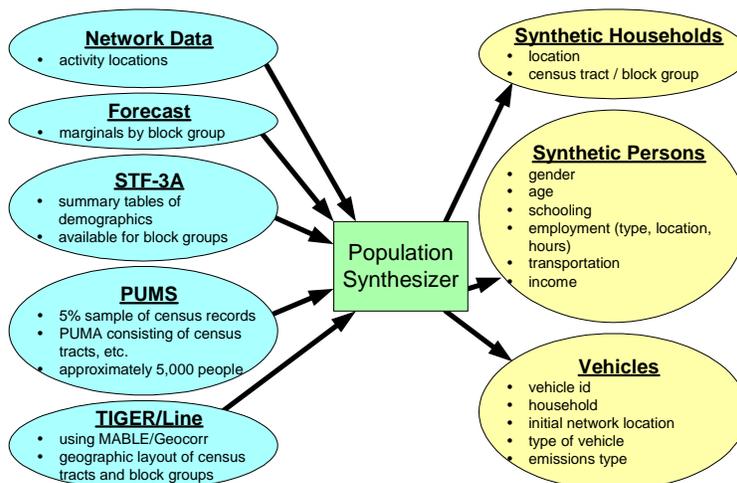


Fig. 1. Data flow for the Population Synthesizer module.

The raw network data were provided by Portland Metro, processed by Los Alamos National Laboratory (LANL), and reviewed by Portland Metro; only activity location data are used during population synthesis. For this, we used the density of residences along streets computed from Portland tax lot data.

Portland Metro provided a 1996 “forecast” (relative to the 1990 census data available for the case study) representing the marginal distribution of the following demographic variables over the 1107 block groups in the study:

- household size
- age of head of household
- annual household income

Although Portland Metro also provided joint distributions of the variables above, they were not used during the population synthesis—we have used them to check the match between the synthetic population and this forecast, however.

The 1990 STF-3A¹ and PUMS² data for the states of Oregon and Washington were obtained directly from the U.S. Census Bureau. We extracted the following STF-3A data fields and the corresponding PUMS fields from these databases:

- family households
 - age of the householder [P24]
 - family income [P107]
 - number of workers in the family [P112]
 - poverty status × race × family type × presence and age of children [P124A,B]
- non-family households
 - household type and gender [P17]
 - race × household type × presence and age of children [P20]
 - age of non-family householder [P24]
 - non-family household income [P110]
 - poverty status × age of householder □ household type [P127]
- group quarters
 - group quarters [P40]
 - group quarters × age [P41]

The MABLE/Geocorr web site³ provided the abstract of the U.S. Census Bureau TIGER/Line data needed for correlating the overlap between block groups in the STF-3A data and PUMAs in the PUMS data.

Finally, Portland Metro provided distributional data for the of 23 CMEM vehicle types⁴, shown in Fig. 2, determined from Oregon motor vehicle records.

¹ Census of Population and Housing, 1990, Summary Tape File 3 on CD-ROM Technical Documentation, prepared by the Bureau of Census.

² Census of Population and Housing, 1990, Summary Tape File 3 on CD-ROM Technical Documentation, prepared by the Bureau of Census.

³ Mable/Geocorr Home Page, <http://plue.sedac.ciesin.org/plue/geocorr>

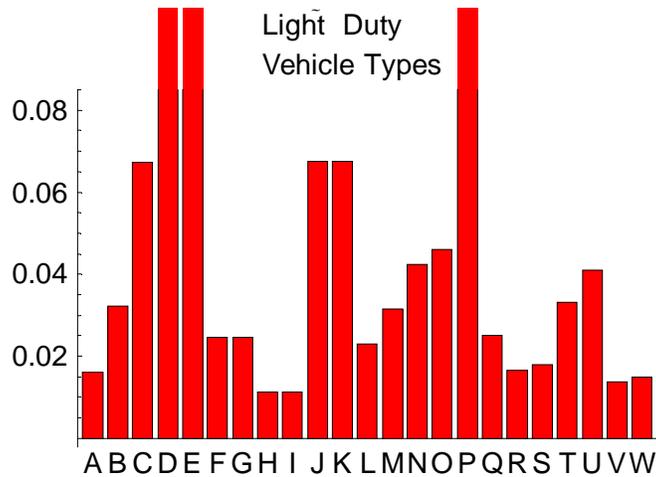


Fig. 2. Distribution of 23 CMEM vehicle types determined from Oregon motor vehicle records.

3. GENERATING THE POPULATION

We generated the synthetic population using the procedure specified in the TRANSIMS Population Synthesizer (see the Population Synthesizer general documentation). Populations for each PUMA of the six PUMAs (01200, 01300, 01400, 01500, 01901, 01902) were generated separately and then merged into a single file, as recommended by the user manual. The contents of the six configuration files used to generate these segments of the population, along with the configuration file used to locate the population and generate the vehicles (*allstr.cfg*) are given in Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Two (*PS-1*) of the Portland Study Reports.

We performed a variety of tests to verify that the Population Synthesizer operated correctly. The straightforward tests of the software included:

- The correct records are extracted from the PUMS data file.
- The correct fields and records are extracted from the STF-3A database.
- The forecast marginal records are correctly read.
- Demographic variables and block group numbers in the above records are correctly indexed.
- The multiway input weights to the “rake” procedure are constructed correctly.
- The marginals in the multiway output tables of the “rake” procedure match those for each block group of the input constraints to “rake” (this applies to the base “rake” run and the forecast “rake” run).

⁴ Barth, M., T. Younglove, T. Wenzel, G. Scora, F. An, M. Ross, and J. Norbeck (1997), “Analysis of Modal Emissions for a Diverse in-use Vehicle Fleet.” *Transportation Research Record, No. 1587*, Transportation Research Board, National Academy of Science, pp. 73-84.

- The iterative proportional fitting procedure in “rake” finds the correct proportion matrix for each block group.
- The correct number of households is generated for each block group.

We performed tests (such as the chi-squared test) on the output to verify the statistical correctness of the results:

- The marginal distributions of the demographics for each block group match the STF-3A and forecast input data for the base and forecast populations, respectively. The results of this chi-square test are shown in Fig. 3.
- The PUMS records drawn for each block group’s base and forecast population match the probability distribution of the “rake” output multiway table.
- The odds ratios in the multiway table fit to the STF-3A matches that for the PUMS data.

We also inspected the located populations and vehicles to verify that:

- Households are placed at activity locations proportionally to area of residences along the street.
- Households are not placed in commercial, industrial, etc., areas.
- Vehicles are assigned according to household demographics.
- Vehicles are parked at the location nearest the household.

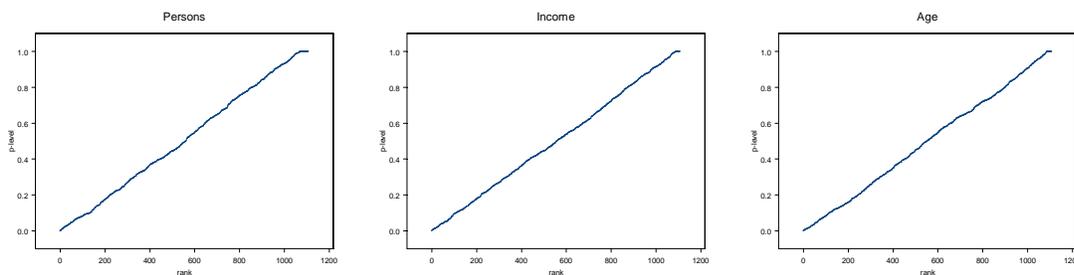


Fig. 3. Probability levels sorted by rank for a chi-squared test of the marginal demographics in 1107 block groups. The strong linearity of the plots indicates a good match to the forecast demographics.

Chapter 3

Activity Generation

1. INTRODUCTION

This chapter documents the input parameters and conditions that were set to generate the initial activity set called AS-1. The general documentation for the Activity Generator may be obtained from Volume 1: TRANSIMS Documentation at <http://transims.tsasa.lanl.gov>. A summary of that document is given here.

The generated population of households and individuals is described in the previous chapter. One synthetic individual is created for each person in the true population. The Activity Generator assigns a set of activities to each synthetic individual in the population. This is done one synthetic household at a time. Based on household demographics, each household in the synthetic population is matched with a collection of households in an activity survey. One of these possible matching households is then selected according to a probability distribution, and the activities of that household are assigned to members of the synthetic household. The sequence of activities in this set is held as constant as possible including all of the intra-household shared rides. The times of the activities (within a range) and the modes to reach the activities also do not change from the survey. The activity locations are, of course, different.

For mode and location choice, the general philosophy for the Activity Generator differs from that of the econometric models under development elsewhere. Those models rely on multi-parameter statistical fits to the survey data. Calibration is accomplished by adding terms to the model until the ground counts and model estimates match to some degree of accuracy. Here the modeling is different. A simple location choice model is fit to the survey, and mode and final location choices are made by a series of feedback loops. (See the red loops in Figure 2, Volume One (*Introduction/Overview*), Chapter Four (*The Portland Study Flow of Runs*) and Volume Three (*Feedback Loops*), Chapter Five (*Mode Choice*) of the Portland Study reports.)

The simple location choice model is

$$P_l = (at)_l \exp(\beta_m (t_{s,l})^k)$$

where

P_l is the probability of choosing location l
 $(at)_l$ is the attractor at location l
 β_m is the mode coefficient for mode m
 $(t_{s,l})^k$ is the travel time between locations s and l
 k is a constant.

The attractor and the mode coefficient in the above equation are determined by a statistical fit to the household survey and the employment and land use data assigned to the activity locations of the network.

Freight and itinerant traveler trips were included in the simulation using trip tables supplied by Portland Metro.

2. SURVEY HOUSEHOLDS

Portland Metro conducted a two-day activity survey of 5863 households. The results of the first day of this survey for 3473 households were used in this study. An example of the raw data from the survey for household 200137 is show below.

2.1.1 RAW SURVEY RECORDS

```

2,4,200137,1,1,1,16,"CHEZ ELLE BEAUTY
SALON",2,900,1,1030,1,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,2,2,,,4,,,850,1,900,1
,0,10,2,"",,1

2,4,200137,1,1,2,20,"HOME",2,1040,1,1120,1,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,3,
2,,,4,,,1030,1,1040,1,0,10,2,"",,1

2,4,200137,1,1,3,14,"MONTGOMERY
WARDS",2,1135,1,1235,2,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,1,2,,,4,,,1120,1,113
0,1,0,10,2,"",,1

2,4,200137,1,1,4,51,"HOME",2,100,2,205,2,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,3,2,
,,4,,,1245,2,100,2,0,15,2,"",,1

2,4,200137,1,1,5,14,"KIENOWS",2,215,2,315,2,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,1,
2,,,4,,,205,2,215,2,0,10,2,"",,1

2,4,200137,1,1,6,51,"HOME",2,325,2,455,2,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,3,2,
,,4,,,320,2,325,2,0,5,2,"",,1

2,4,200137,1,1,7,11,"PANCHOS",2,500,2,600,2,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,2,2
,2,,,4,,,455,2,500,2,0,5,2,"",,1

2,4,200137,1,1,8,51,"HOME",1,605,2,100,1,2,"",,,,,,,,,,"",,,,,,,,,,"",,,,,,,,,,
,,,,,"",,1

2,4,200137,1,2,1,12,"GOOD SAMARITIAN
HOSPITAL",2,215,2,1130,2,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,1,2,,,4,,,200,2,21
5,2,0,15,2,"",,1

2,4,200137,1,2,2,51,"HOME",2,1145,2,200,1,2,7,"",,,,,,,,,,"",,,,,,,,,,"",1,1,1,3,2
,,4,,,1130,2,1145,2,0,15,2,"",,1
  
```

```

2,4,200137,2,1,1,51,"HOME",1,830,1,450,2,2,,,"",,,,,,,,,,"","",,,,,,,,,,"",,,,,,,,,,
,,,,,"",,1

2,4,200137,2,1,2,11,"PANCHOS",2,500,2,600,2,2,7,"",,,,,,,,,,"","",,,,,,,,,,"",,1,2,2,2
,2,,,4,,,455,2,500,2,0,5,2,"",,1

2,4,200137,2,2,1,51,"HOME",1,830,1,100,1,2,,,"",,,,,,,,,,"","",,,,,,,,,,"",,,,,,,,,,
,,,,,"",,1

```

In TRANSIMS, the survey results are transformed to a simple file giving the person and household IDs, the activity types, the mode of travel, the start and end times of the activities, and the locations where the activity took place. An example of the entries in the file for household 200137 is given in Table 1.

Table 1. The raw activity survey data is transformed to a file with a minimum of entries. This table shows the entries in the file for household 200137. SAMPNO is the household number. The person ID is given in the PERSNO field. The activities are numbered sequentially in ACTNO. Each activity type is given a number identifier and is in the column marked ACTID. A flag in the AT_HOME column denotes whether the activity took place at home. WRTHERE is a flag that denotes a change in activity locations. Mode and driver are given in their respective columns. NUMVEH is the number in the vehicle. The location of the activity is geo-coded and given in columns GEOX and GEOY.

SAMPNO	PERSNO	ACTNO	ACTID	AT_HOME	WRTHERE	MODE	DRIVER	NUMVEH	ACTSTART	ACTEND	GEOX	GEOY
200137	1	0	0	1	2	1	0	0	0	540	7657414	686984.3
200137	1	1	3	2	2	2	1	1	540	630	7653721	688412.6
200137	1	2	2	1	2	2	1	1	640	680	7657414	686984.3
200137	1	3	2	2	2	2	1	1	695	755	7673690	681890.1
200137	1	4	0	1	2	2	1	1	780	845	7657414	686984.3
200137	1	5	2	2	2	2	1	1	855	915	7656295	688683.8
200137	1	6	0	1	2	2	1	1	925	1015	7657414	686984.3
200137	1	7	4	2	2	2	1	2	1020	1080	7656613	687307.8
200137	1	8	0	1	2	2	1	2	1085	1500	7657414	686984.3
200137	1	9	0	1	1	1	0	0	1500	1620	7657414	686984.3
200137	2	0	0	1	2	1	0	0	0	510	7657414	686984.3
200137	2	1	0	1	1	1	0	0	510	1010	7657414	686984.3
200137	2	2	4	2	2	2	2	2	1020	1080	7656613	687307.8
200137	2	3	0	1	2	2	2	2	1085	1620	7657414	686984.3

The complete TRANSIMS activity file contains 13 columns. These are:

- 1) SAMPNO: This column is the household ID from the survey.
- 2) PERSNO: The person ID from the survey is in this column.
- 3) ACTNO: The activities for each person are number sequentially and given here.
- 4) ACTID: The survey contains 28 activity types. With the concurrence of Portland Metro, these were collated into nine activity types for this study. The nine activity types are given in
- 5)
- 6) Table 2. School or college activity types in the table were determined by age of the participant.

Table 2. This table gives the TRANSIMS activity IDs used in the study.

TRANSIMS ID	TRANSIMS Description
0	Home
1	Work
2	Shop
3	Visit
4	Social/Recreational
5	Other
6	Serve Passenger
7	School
8	College

The correspondence between the activity types of the survey and those used in the study is in Table 3.

Table 3. This table gives the correspondence between the 28 activity types of the survey and the nine activity types used in the study.

Survey ID	Survey Description	TRANSIMS ID	TRANSIMS Description
11	Meals	4	Social/Recreational
12	Work	1	Work
13	Work-related	1	Work
14	Shopping (general)	2	Shop
15	Shopping (major)	2	Shop
16	Personal services	3	Visit
17	Medical care	3	Visit
18	Professional services	3	Visit
19	Household or personal business	3	Visit
20	Household maintenance	2	Shop
21	Household obligations	3	Visit
22	Pick-Up/Drop-Off passengers	6	Serve Passenger
31	Visiting	4	Social/Recreational
32	Casual entertaining	4	Social/Recreational
33	Formal entertaining	4	Social/Recreational
41	School	7or 8	School/College
42	Culture	4	Social/Recreational
43	Religion/Civil Services	4	Social/Recreational
44	Civic	4	Social/Recreational
45	Volunteer work	4	Social/Recreational
51	Amusements (at-home)	0	Home
52	Amusements (out-of-home)	4	Social/Recreational
53	Hobbies	4	Social/Recreational
54	Exercise/Athletics	4	Social/Recreational
55	Rest and relaxation	4	Social/Recreational
56	Spectator athletic events	4	Social/Recreational
90	Incidental trip	5	Other
91	Tag along trip	5	Other

The anchor activities (see the general documentation for the Activity Generator) were home, work, and school or college. The Route Planner time priorities (see the general documentation for the Route Planner) for each activity type are given in the configuration file in Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Three (*AS-1*) of the Portland Study Reports.

- 7) AT_HOME: This is a flag to show whether the activity took place at home. A “1” in the column indicates that it was a home activity.
- 8) WRTHERE: This column is used by the Activity Generator to determine if a trip was made between the activity and the preceding one. A “1” in the column indicates no change in activity locations.
- 9) MODE: The modes’ IDs from the raw survey were transformed to nine modes used in this study. In TRANSIMS, each mode is given a symbol for use in the Route Planner. The mode numbers, symbols, and their meanings are in Table 4.

Table 4. The mode numbers, symbols, or mode strings, and a description of the modes used in the Study are given here. Each mode string starts and ends with the symbol “w”, which indicates a walk.

Number	Symbol	Meaning
1	w	walk
2	wcw	walk-car-walk
3	wtw	walk-transit-walk
4	wlw	walk-light rail-walk
5	wcwtw	park&ride in
6	wtwcw	park&ride out
7	wiw	bike
8	wkw	intra-household shared ride
9	wKw	school bus

Each TRANSIMS activity takes place at an activity location. Parking locations, where vehicles start and end their journeys, are different from activity location. To access a vehicle, a traveler must walk between the activity location and the parking location.

- 10) DRIVER: This is an indicator variable to establish the driver for an automobile trip. A “1” in this column indicates the driver of the vehicle.
- 11) NUMVEH: This variable is the number of travelers in the vehicle on the trip.
- 12) ACTSTART: This is the start time of the activity in minutes since midnight.
- 13) ACTEND: This is the end time of the activity in minutes since midnight.
- 14) GEOX and GEOY: These two columns are the geocoded locations of the survey activities.

Many of the reported survey activities and times were edited to make them consistent with TRANSIMS. In particular, the times activities took place were changed when there were obvious mismatches in the times for shared rides.

Table 5 demonstrates one such survey result. The activities are for the two persons in survey household 200137. The table shows the household ID (*hhid*), the activity types (*i1* and *i2*), the at-home flags (*h1* and *h2*), the were-there flags (*w1* and *w2*), the modes (*m1* and *m2*), the driver flags (*d1* and *d2*), and the number of persons on the trip (*p1* and *p2*). The last three time periods are the ones of interest. At 1020, the two persons arrive together at an activity in a car with the first person as the driver. They then return home with the first person in the household the driver. The highlighted lines in the table show the return to home. In the survey, the first person reported the return to home at 1085, while the second person reported the return to be at 1080. For TRANSIMS, these two times were made to match.

Table 5. The survey data for household 200137 has an obvious mismatch in the shared ride times for the last activity for both travelers in the household. The activity after the last shared ride is scheduled to begin at 1085 for the first person in the household and at 1080 for the second person.

hhid	time	i1	h1	w1	m1	d1	p1	i2	h2	w2	m2	d2	p2
200137	0	0	1	1	-1	-1	-1	0	1	1	-1	-1	-1
200137	540	5	2	2	2	1	1	x	x	x	x	x	x
200137	640	5	1	2	2	1	1	x	x	x	x	x	x
200137	695	2	2	2	2	1	1	x	x	x	x	x	x
200137	780	0	1	2	2	1	1	x	x	x	x	x	x
200137	855	2	2	2	2	1	1	x	x	x	x	x	x
200137	925	0	1	2	2	1	1	x	x	x	x	x	x
200137	1020	5	2	2	2	1	2	5	2	2	2	2	2
200137	1080	x	x	x	x	x	x	0	1	2	2	2	2
200137	1085	0	1	2	2	1	2	x	x	x	x	x	x

3. BINARY TREE

In TRANSIMS, each household in the synthetic population is assigned the activity sequence from one of the survey households. These are assigned by matching some of the household demographics in the survey and the synthetic population. Since there are not enough survey households to do an exhaustive cross-classification of potentially useful demographics for this match, a classification tree is used for the matching. The procedures for generating such a tree are given in the general documentation on the Activity Generator.

The classification tree used in this study is given in Table 6. The classification tree itself is partially in table form. The most important variables make up the “row” and “column” of the table. These are the number of workers (*WORKERS*) and the household size (*HHSIZE*). Other variables in the tree are: age of the householder (*HHAGE*), the household income grouped in categories (*INCOME*), the number of persons in the household less than 5 years of age (*ALT5*) and between 5 and 15 years of age (*A5-15*), and the household density in households/acre at the home location (*HHDensity*).

The number of survey households at each node in the tree is given in parentheses. These are further classified as those survey households with no non-driving children or with children less

than 16 years of age. It is interesting to note the disproportionate number of older single- or two-person households with no workers in the survey. These represent retired persons who have time to fill out the surveys. This points out a general need to unbiased the survey results for the census. By taking into account the demographics of both the synthetic population and the survey population, the activity generation in TRANSIMS does this automatically.

Table 6. The classification tree for assignment of survey households is partially in table form. The two most important classification demographics are the number of workers in the household and the household size (HHSIZE). The number of survey households in each category are given in parentheses.

WORKERS=0	WORKERS=1	WORKERS=2	WORKERS>2
HHSIZE=1 HHAGE<38.5 (34)(34,0) HHAGE>38.5 (319)(319,0)	HHSIZE=1 (705)(705,0)	HHSIZE=2 (677)(677,0)	HHSIZE=3 (75)(75,0)
HHSIZE=2 HHAGE<53.5 INCOME<5.5 (25)(12,13) INCOME>5.5 (15)(12,3) HHAGE>53.5 (268)(267,1)	HHSIZE=2 A5-15=0 (304)(291,13) A5-15>0 (48)(0,48)	HHSIZE=3 (248)(78,170)	HHSIZE=4 HHAGE<46.5 (28)(17,11) HHAGE>46.5 (24)(18,6)
HHSIZE>2 (32)(8,24)	HHSIZE=3 A5-15<=1 HHAGE<29.5 (18)(1,17) HHAGE>29.5 (89)(36,53) A5-15>1 (24)(0,24)	HHSIZE=4 ALT5=0 A26-45=0 (21)(8,13) A26-45>0 (134)(7,127) ALT5>0 (69)(0,69)	HHSIZE=4 (43) (10,33)
	HHSIZE=4 A5-15<=1 (73)(4,69) A5-15>1 (49)(0,47)	HHSIZE=4 A5-15<=2 (49)(1,48) A5-15>2 HHDensity<1.29 (18)(0,18) HHDensity>1.29 (19)(0,19)	
	HHSIZE=4 A5-15<=2 (40)(1,39) A5-15>2 (27)(0,27)		

KEY
(32)(8,24)
32 households
8 with no nondriving children
24 with nondriving children

4. WEIGHTING THE SYNTHETIC HOUSEHOLDS

In TRANSIMS, an activity pattern for a synthetic household is determined by overlaying an activity pattern from a similar household from the household survey. Each synthetic household in the TRANSIMS population is assigned one of the activity patterns of the survey households at a node in the tree. The survey household is chosen randomly in proportion to the weight assigned to the survey household.

In this methodology, each survey household is assigned a weight. The weights assigned to the survey households at a node in the tree are proportional to the probability of selecting that household's activity sequence. Each survey household at a node of the tree has an equal chance of being selected if the weights of all households at the node are unity.

The first step in generating an activity set for the synthetic population is to let all of the survey households have unit weights. Because of underreporting of trips in the survey households or sampling households with less active lifestyles, the total number of trips generated for the synthetic population using this procedure is usually too low. To remedy this, the survey households with more trips are weighted higher than those with less trips. The weighting scheme used in the Portland Study is described below.

Suppose that with unit weights, there are $p\%$ too few trips. (In the Portland Study, $p=25\%$ for combined drive and transit trips). Let $X = (1 + p / 100)$. At a node in the tree, let the number of drive and transit trips for the m survey households at the node be

$$n_1, n_2 \dots n_m$$

If each survey household at the node has a unit weight for selection to overlay its activity pattern on a synthetic household, then the average number of drive plus transit trips generated for the synthetic households at this node is

$$\bar{n} = \sum n_i / m$$

The goal is to find a set of weights, w_1, w_2, \dots, w_m so that the weighted average of the number of trips generated at a node is X times the original average. That is

$$\sum w_i n_i / \sum w_i = X \bar{n}$$

Since $X > 1$, this equation may be satisfied by giving the survey households with more than the average number of trips more weight. Here we use

$$w_i = (1 + \beta n_i^k)$$

as the general form of the weighting function, where β and k are calibration constants.

The form of this function for various values of k is shown in *Fig. 1*. The lines in this figure represent the relative weights from $k=0$ to $k=6$, based on the number of trips in the household. Those lines with the smallest slopes give more similar weighting to all of the households at the

node of the tree. It may not be desirable to assign one household significantly more weight than all of the others. In these cases, the activity pattern of the one household would be overutilized at the expense of the others. Given the steep slopes of these lines for the larger values of k , it is probably best to use the smallest k that satisfies the equation above.

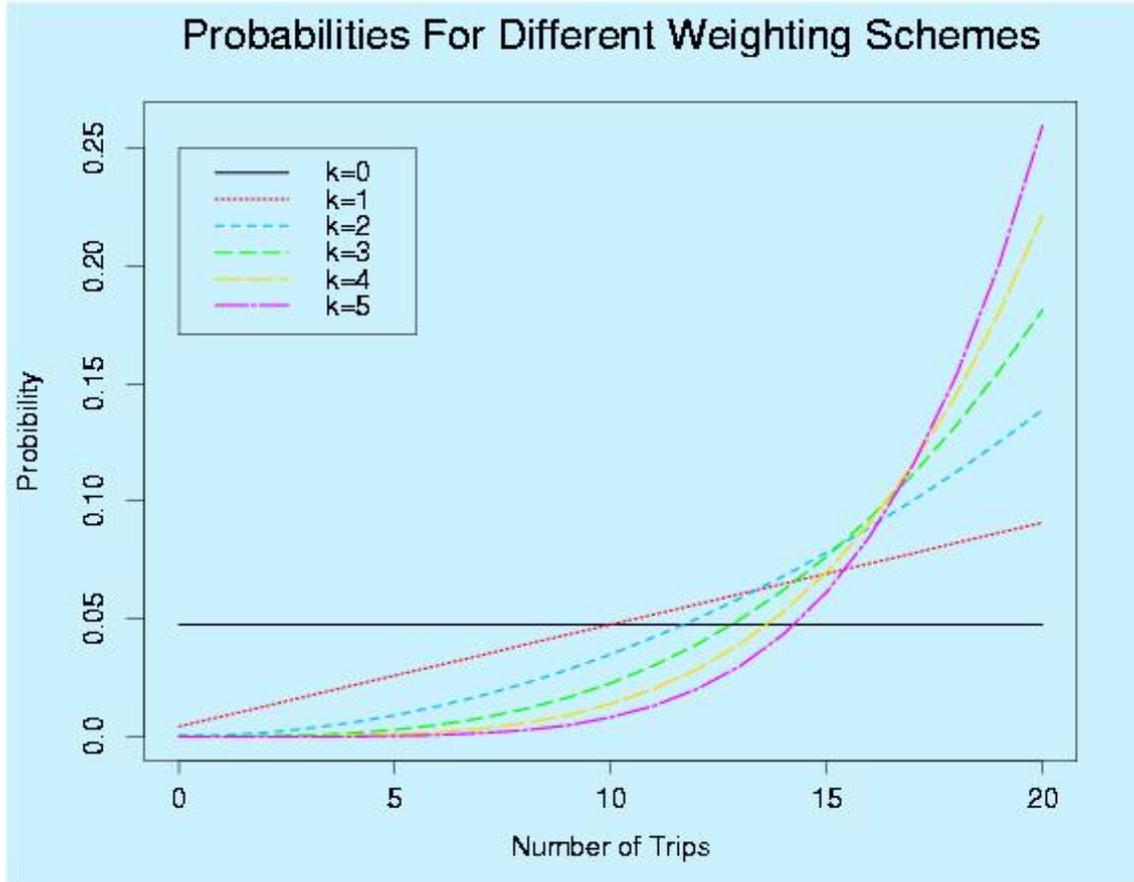


Fig. 1. The probability of selecting households given the number of transit and auto trips as the weighting value, k , changes.

In the Study, the smallest integer value of k was determined for each node so that the weights remained positive. β is obtained by solving

$$\sum (1 + \beta n_i^k) n_i / \sum (1 + \beta n_i^k) = X\bar{n}$$

The solution for β in this equation is

$$\beta = (X - 1)m\bar{n} / [\sum n_i^{k+1} - X\bar{n} \sum n_i^k]$$

For a particular value of k , the solution for β in the above equation may be less than 1. For these cases, a larger value of k must be selected. Solutions of the equation with negative values of β lead to negative weights for some of the households. The solutions for k for each node in the classification tree are given in Table 7.

Table 7. The values of k for each of the 45 nodes in the classification tree. In some cases, where there is a small number of households at the node, no solution for k exists, so the value of 0 was used.

DEMO-1	DEMO-2	DEMO-3	DEMO-4	CHILDREN	#SURVEY	K
Workers=0	HHSIZE=1	HHAGE<38.5		No	34	1
Workers=0	HHSIZE=1	HHAGE>38.5		No	319	1
Workers=0	HHSIZE=2	HHAGE>53.5		No	267	2
Workers=0	HHSIZE=2	HHAGE<53.5	INCOME<5.5	No	12	1
Workers=0	HHSIZE=2	HHAGE<53.5	INCOME>5.5	No	12	1
Workers=0	HHSIZE=2	HHAGE>53.5		Yes	1	0
Workers=0	HHSIZE=2	HHAGE<53.5	INCOME<5.5	Yes	13	1
Workers=0	HHSIZE=2	HHAGE<53.5	INCOME>5.5	Yes	3	0
Workers=0	HHSIZE>2			No	8	2
Workers=0	HHSIZE>2			Yes	24	1
Workers=1	HHSIZE=1			No	705	1
Workers=1	HHSIZE=2	A5-15=0		No	291	1
Workers=1	HHSIZE=2	A5-15=0		Yes	13	2
Workers=1	HHSIZE=2	A5-15>0		Yes	48	1
Workers=1	HHSIZE=3	A5-15<=1	HHAGE<29.5	No	1	0
Workers=1	HHSIZE=3	A5-15<=1	HHAGE>29.5	No	36	1
Workers=1	HHSIZE=3	A5-15>1		yes	24	2
Workers=1	HHSIZE=3	A5-15<=1	HHAGE<29.5	Yes	17	1
Workers=1	HHSIZE=3	A5-15<=1	HHAGE>29.5	Yes	53	1
Workers=1	HHSIZE=4	A5-15<=1		No	4	1
Workers=1	HHSIZE=4	A5-15<=1		Yes	69	1
Workers=1	HHSIZE=4	A5-15>1		Yes	47	1
Workers=1	HHSIZE>4	A5-15<=2		No	1	0
Workers=1	HHSIZE>4	A5-15<=2		Yes	39	2
Workers=1	HHSIZE>4	A5-15>2		Yes	27	2
Workers=2	HHSIZE=2			No	677	2
Workers=2	HHSIZE=3			No	78	2
Workers=2	HHSIZE=3			Yes	170	2
Workers=2	HHSIZE=4	ALT5=0	A26-45=0	No	8	2
Workers=2	HHSIZE=4	ALT5=0	A26-45>0	No	7	1
Workers=2	HHSIZE=4	ALT5>0		Yes	69	1
Workers=2	HHSIZE=4	ALT5=0	A26-45=0	Yes	13	2
Workers=2	HHSIZE=4	ALT5=0	A26-45>0	Yes	127	2
Workers=2	HHSIZE>4	A5-15<=2		No	1	0
Workers=2	HHSIZE>4	A5-15<=2		Yes	48	2
Workers=2	HHSIZE>4	A5-15>2	HHdensity<1.29	Yes	18	1
Workers=2	HHSIZE>4	A5-15>2	HHdensity<1.29	Yes	19	2
Workers>2	HHSIZE=3			No	75	2
Workers>2	HHSIZE=4	HHAGE<46.5		No	17	2
Workers>2	HHSIZE=4	HHAGE>46.5		No	18	3
Workers>2	HHSIZE=4	HHAGE<46.5		Yes	11	4
Workers>2	HHSIZE=4	HHAGE>46.5		Yes	6	3
Workers>2	HHSIZE>4			No	10	4
Workers>2	HHSIZE>4			Yes	33	3

These are the minimum integer values of k that assign positive weights to each household in the node. Using these values of k and the fit value of β , each household was assigned the weight

$$w_i = (1 + \beta n_i^k)$$

5. ACTIVITY LOCATION ATTRACTORS/LAND USE

Activity attractors were computed for each of the eight activity types (home is excluded) in Table 8 and placed at each activity location. The attractors are a function of “land-use” characteristics at each of the activity locations in the network. The “land-use” data includes total employment, the number of households, total college enrollment, service employment, retail employment, other employment, government employment, and park area in acres. These data were combined using the functional form:

$$\prod_i a_i^{k_i^{(j)}}$$

where a_i is “land-use” data i , and $k_i^{(j)}$ is an estimated coefficient for travel mode j .

Table 8. Land-use data that include employment numbers by employment types are combined by the formula $\prod_i a_i^{k_i^{(j)}}$ to make attractors for each activity type in the Study.

Activity Type	Land-Use 1	Land-Use 2	Land-Use 3	Land-Use 4	Land-Use 5
Work	Total employment				
Shop	Number Households	Service Employment	Retail Employment	Other Employment	
Visit	Number Households	Service Employment	Retail Employment	Other Employment	
Social/Recreational	Number Households	Retail Employment	Park Area (in acres)		
Other	Number Households	Service Employment	Retail Employment	Other Employment	Government Employment
Serve Passenger	Number Households	Total employment			
School		NA			
College	Total Enrollment				

In the generation of AS-1, activity locations were assigned the attractor values directly from the application of the estimated formula:

$$\prod_i a_i^{k_i^{(j)}}$$

The value of zero was assigned to the activity location if all of the “Land-Use” variables in Table 8 had the value of zero at the location. In some of the feedback studies, these zero attractors were given a small value greater than zero. This allowed activity locations to be moved to any location.

6. MODE COEFFICIENTS

Location choice in the Study and in TRANSIMS in general is made using a simple logistic model. The form of the function is

$$p_j(i, \beta_m, A_j, K_j, t_{i,j}) \propto \prod_l a_l^{k_l^{(j)}} \exp(\beta_m t_{i,j}^{1/2})$$

where

- the origin is location i ,
- the potential destination is j ,
- β_m is a coefficient that is estimated for each mode m ,
- A_j is the set of land-use variables a_l ,
- K_j is a set of estimated coefficients $k_l^{(j)}$, and
- $t_{i,j}$ is the travel time from location i to location j .

The coefficients β_m and $k_l^{(j)}$ were estimated by maximum likelihood using the survey data as a base. Travel times were taken as scaled distances, and the locations were the 1260 travel analysis zones used by Portland Metro. The statistical fit was better using the square root of the travel time rather than the travel time itself, so the square root was used in the Study.

The coefficients, β_m , for each activity type and mode are shown in Table 9. Some of the coefficients in the table were not estimated because of a lack of data or the manner in which TRANSIMS treats some of the activities and modes, for example schools. Mode coefficients for park and ride (modes 5 and 6) were estimated as a combination of auto and transit (modes 2 and 3). In all cases, data for modes transit (3) and Max (4) were combined and a single estimate was used for both. Coefficients for activity type home (0) were taken to be the same as the work estimates.

Table 9. The coefficients, β_m , for each activity type and mode.

Coefficient	Activity Type	Mode	Coefficient	Activity Type	Mode
-0.129	0	1	-0.110	4	6
-0.057	0	2	-0.134	4	7
-0.098	0	3	-0.125	4	8
-0.098	0	4	-0.125	4	9
-0.080	0	5	-0.493	5	1
-0.080	0	6	-0.384	5	2
-0.098	0	7	-0.450	5	3
-0.057	0	8	-0.450	5	4
-0.057	0	9	-0.450	5	5
-0.129	1	1	-0.450	5	6
-0.057	1	2	-0.490	5	7
-0.098	1	3	-0.384	5	8
-0.098	1	4	-0.384	5	9
-0.080	1	5	-0.517	6	1
-0.080	1	6	-0.163	6	2
-0.098	1	7	-0.300	6	3
-0.057	1	8	-0.300	6	4
-0.057	1	9	-0.300	6	5
-0.166	2	1	-0.300	6	6
-0.159	2	2	-0.500	6	7
-0.121	2	3	-0.163	6	8
-0.121	2	4	-0.163	6	9
-0.140	2	5	-0.517	7	1
-0.140	2	6	-0.160	7	2
-0.100	2	7	-0.300	7	3
-0.121	2	8	-0.300	7	4
-0.121	2	9	-0.300	7	5
-0.087	3	1	-0.300	7	6
-0.138	3	2	-0.500	7	7
-0.117	3	3	-0.160	7	8
-0.177	3	4	-0.160	7	9
-0.120	3	5	-0.129	8	1
-0.120	3	6	-0.057	8	2
-1.610	3	7	-0.098	8	3
-0.138	3	8	-0.098	8	4
-0.138	3	9	-0.080	8	5
-0.142	4	1	-0.080	8	6
-0.125	4	2	-0.098	8	7
-0.092	4	3	-0.057	8	8
-0.092	4	4	-0.057	8	9
-0.110	4	5			

7. ITINERANT TRAVELERS, FREIGHT, AND TRAVEL TIMES

Portland Metro supplied numerous existing data sets to facilitate this demonstration of TRANSIMS technology. Itinerant travelers and freight, but not local deliveries, were generated using trip tables for the 1260 travel analysis zones and the Trip Table Activity Generator (see the general documentation on the Activity Generator for information on this procedure). A time of day is associated with each trip table. Additional data from Portland Metro included zone-to-zone travel times by time of day for drives, transit, and park and ride.

Zone-to-zone trip tables representing freight trips were given for four times of day, all day, 14:00-15:00, 16:30-17:30, and 15:30 to 17:30. These were converted to one trip table and the times for the trip were in proportion to the trips in each of these tables. The Trip Table Generator then produced one trip for each of the trips in the table. The endpoints of the trips were determined by the values of the attractors, T_{Org} and T_{Des} , on the network activity locations.

Auto trips representing itinerant travelers were generated by the Trip Table Activity Generator from two trip tables supplied by Portland Metro. The four tables represent four times of day: (7:00-9:00), (9:00-16:00), (16:00-18:00), and (0:00-7:00, 18:00-24:00). The time file developed for the Trip Table Generator for these trips reflects these times.

Portland Metro supplied zone-to-zone travel times by mode and time of day in five tables. These tables are for a.m. two-hour auto, a.m. two-hour total transit, a.m. two-hour park and ride times, mid-day one-hour auto travel times, and one-hour mid-day total transit travel times. These tables were used to construct travel times for each mode for 24 hours. The method was:

1) Walk

The auto mid-day travel times multiplied by 40 mph were used for all walk times.

2) Drive (Auto)

The following combinations of travel times were used:

- a) 0 to 6:30 a.m.: mid-day travel times multiplied by .9 were used.
- b) 6:30 a.m. to 9:00 a.m.: the a.m. two-hour travel times were used.
- c) 9:00 a.m. to 4:30 p.m.: mid-day travel times were used.
- d) 4:30 p.m. to 6:30 p.m.: the a.m. two-hour travel times were used with the origin and destination zones reversed.
- e) 6:30 to midnight: mid-day travel times multiplied by .9 were used.

3) Transit

The transit trip tables were treated the same as the auto trip tables.

4) Light Rail

Transit tables were used. This choice has implications in the mode selection feedback loop. In essence, since the Route Planner will find the best transit path between bus and light rail, this mode is probably not necessary.

5) Park and Ride

The one park and ride travel time table was used for the entire 24 hours.

- 6) Bike
The walk travel times were divided by 10 and were used as bike travel times.
- 7) Inter-household shared ride
Auto travel times were used for this mode.
- 8) School bus
Auto travel times were used for this mode.

The Activity Generator has the ability to use the travel time files directly, and this is recommended if there are many tens or hundreds of thousands of zones. Here, however, the number of zones is 1260, and the travel time tables fit into the memory of the machine. A special travel time function was coded into the Activity Generator and Activity Regenerator to read and store these tables. Use of the special travel time function improved the execution speed of the Activity Generator dramatically and provided a mechanism to manipulate travel times for special conditions during feedback loops.

8. RUN PARAMETERS (CONFIGURATION FILE KEYS)

The configuration file keys used to generate activity set AS-1 are given in Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Three (*AS-1*) of the Portland Study Report series. AS-1 is the building block for all of the activity sets that follow. Particular attention was paid to the number of drive and transit trips. These were calibrated to the number that Portland Metro thought necessary by weighting the selection of the survey households. Configuration files used to generate trips for freight and itinerant travelers are also included in Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Functions*), Chapter Three (*AS-1*) of the Portland Study Report series.

Chapter 4

Router

1. INTRODUCTION

This chapter documents the input parameters and conditions that were set to generate the initial route set called RS-7. It was generated from activity set AS-7. The general documentation may be obtained from Volume 1: TRANSIMS Documentation available at <http://transims.tsasa.lanl.gov>. A summary of that document is given here.

Three sets of activities were routed separately: population (AS-7), trucks, and itinerants. The itinerants were further divided into four groups by time of day: a.m., mid-day, p.m., and rest of the day.

The activities generated for the population, trucks, and itinerants are described in the previous chapter. For each pair of successive activities for a given traveler which takes place in different locations, a route through the network is computed. This path is close to the shortest path through the network using the mode specified in the activity file. The section below on link delay noise explains why the path may not be the shortest. The possibly multi-modal path is then divided into unimodal legs and written to the plan file. For each trip, the plan file contains the starting and ending locations, the start time and expected duration of the trip, the distance traveled, and mode-specific path information, such as the nodes passed through for auto trips or route numbers for transit trips.

2. TRAVEL MODES

There are a total of nine transportation modes used in AS-7: auto, walk, bike, transit, light rail, school bus, inter-household shared ride, and park and ride (outgoing and return).

Auto mode (wcv) is used for vehicle trips between activity locations. It is routed as two separate pieces. The first is a walk trip from the source activity location to the current location of the vehicle to be used for the trip (the source parking location). The location of the vehicle is remembered, so that successive auto trips using the same vehicle will use the correct parking location. The second part of the trip is an auto trip from the source parking location to the destination parking location, followed by a walk trip from the destination parking location to the destination activity location. The destination parking location is a parking location connected by a process link to the destination activity location.

Walk mode (w) and bike mode (wiw) are used for foot or bike traffic between two activity locations. They are routed identically with the exception of the speed used—one meter per second for walk mode, and five meters per second for bike mode.

Transit mode (wtw) trips can use buses or light rail for travel between activity locations. Travelers are allowed to transfer between transit routes at a particular stop and to walk between transit stops. Travelers may also walk from the source activity location to the destination activity location if that is faster than using transit vehicles. In order to reduce the number of travelers making transfers that save only a few seconds, a time penalty of 120 seconds is imposed for each transit vehicle boarded. Light rail mode (wlw) is similar to transit mode, with the exception that only light rail routes may be used.

Park and ride has two modes—outbound (wcwtw) and return (wtwcw). Outbound is used for trips leaving home, and return for trips returning to home. An outbound trip is routed similarly to auto, with a walk to the location of the vehicle, followed by a trip that included a drive portion to a park-and-ride parking location, followed by a transit trip to the destination activity location. The transit portion is the same as a transit trip with regard to transfers and walking. A return trip is routed as a transit trip to the location of the vehicle, followed by an auto trip to the destination activity location.

There are two unrouted modes—school bus (wkw) and inter-household shared ride passenger (wKw). Trips of these types are collectively known as “magic move” trips because they are not routed through the transportation network. Instead, the duration is determined from the times of the activities before and after the trip. The Traffic Microsimulator simply moves the travelers from one location to the other.

3. LINK DELAYS

The link delays used for the creation of RS-7 were free-speed delays; that is, the speed at which vehicles would be expected to travel without interference from other vehicles on the road. Starting with RS-8, link delays generated by the Traffic Microsimulator were used.

These delays are the average delay experienced by all of the vehicles on a particular link exiting the link in a 15-minute period. The Route Planner uses these data points, along with the free speed on the link at simulation time 0 and 24 hours, and interpolates between them to generate the delay on a particular link at a particular time.

The problem with this approach is that some links may become completely jammed in a short period of time so that no vehicles exit the link and no data is output by the Traffic Microsimulator. The interpolation by the Route Planner then produces link delay times that are artificially low. To remedy this, the microsimulation configuration file was adjusted to create another output file that lists the links for which no vehicles exited the link during a particular 15-minute interval, but for which there were vehicles present on the link. This allows jammed links to be distinguished from empty links. Link delays were then added for these particular links and time intervals that are the equivalent of the vehicles traveling 0.5 meters per second.

4. OTHER PARAMETERS

4.1 Overdo

Overdo is a heuristic that is intended to reduce the number of possible paths explored by the Route Planner, thus reducing its execution time. The normal search is similar to a circle centered on the source activity location. The size of the circle increases until the destination activity location is found within the circle, as shown in Fig. 1. The overdo parameter changes the shape of the circle to an ellipse, with the long axis of the ellipse aligned with a line connecting the source and destination activity locations, as shown in Fig. 2. The value of the overdo parameter controls the width of the ellipse.

This method works well, in general. However, some geometric configurations in the network cause incorrect paths to be produced. Because of this problem, the overdo parameter was not used for the Portland Study.

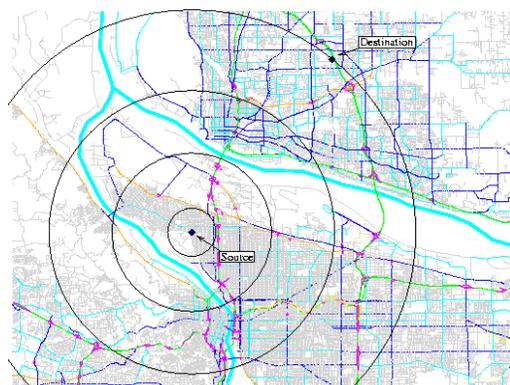


Fig. 1. Search without the overdo parameter.

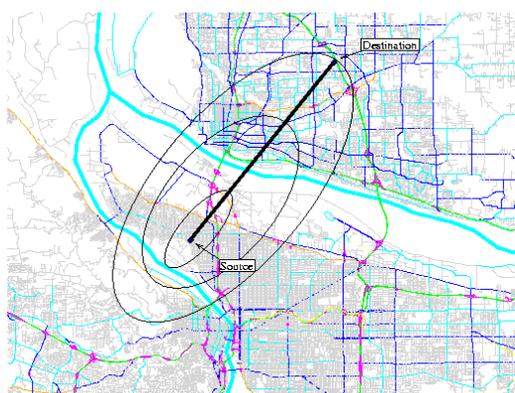


Fig. 2. Search with the overdo parameter.

4.2 Link Delay Noise

Normally, the Route Planner chooses the shortest route between two activity locations. However, there are usually additional paths that are only slightly longer which are viable alternatives. These paths are considered by adding noise to the link delays, so that additional paths are considered.

The link delay noise parameter is specified as a fraction of the link delay to add or subtract. If the delay on a link is d seconds, and the noise factor is f , then the delay used is selected from the uniform distribution $(d-(d*f), d+(d*f))$. A delay noise factor of 0.15 was used to route the population and itinerants, while 0 was used to route the trucks. This ensured that trucks always travel by freeway, which is the shortest path.

4.3 Maximum Leg Time

The maximum leg time parameter is the maximum travel time allowed for a particular leg. While routing, if the current trip exceeds this time, the trip is canceled and an anomaly is reported. For the population, a setting of 10800 seconds (three hours) was used. For trucks and itinerants, a setting of 72000 seconds (20 hours) was used. The large value for trucks and itinerants is used because these trips will never be rerouted during feedback, so it is important to route as many as possible.

4.4 Maximum Trip Time

The maximum trip time parameter limits the start time of a trip. This is done to skip routing of any trips that will not be simulated. A value of 97200 seconds (27 hours) was used for all Portland runs.

5. EXECUTION

The Route Planner is both threaded and distributed. For the Portland Study, AS-7 was routed on 60 nodes. Each node was comprised of two 500 MHz Pentium III processors and 1 Gb of memory. A separate Route Planner process was started on each node. Each process used two routing threads and one I/O thread. The hardware is unstable at times, and 10% to 15% of the nodes would die before the routing was completed—usually during the initialization phase. To deal with this situation, restart functionality was added to the Route Planner.

Each process is independent of the others and has a list of households to route. After each household is routed, its ID is written to the completed household file. If the process crashes after routing has started, the *CreatePartialHHFile* script is run to create a new list of households. This list contains only those households that have not yet been routed. This new household file is used to restart the Route Planner from (approximately) where it left off.

After a router has been restarted, two plan files exist: one from the original router process and one from the restarted router process. If the original router process dies after writing some plans to the plan file, but before completion, the household ID is not written to the completed household file. This may lead to identical plans in both plan files and, in the worst case, an

incomplete plan in the original plan file. The *CleanPlan* script removes these duplicate plans and merges the two plan files into one complete plan file.

Chapter 5

Microsimulation

1. INTRODUCTION

This section documents the input parameters and conditions that were set to generate travel times from plan files using the Traffic Microsimulator. The general documentation for the Traffic Microsimulator may be obtained from Volume 1: TRANSIMS Documentation available at <http://transims.tsasa.lanl.gov>. A summary of that document is given here.

The detailed daily plans for each person in the population, including truck drivers and other transients, are generated by the Route Planner as described in the previous chapter. The Traffic Microsimulator executes all of these plans simultaneously on the transportation infrastructure, taking into account interactions between vehicles. In particular, the Traffic Microsimulator models realistically such congestion effects as:

- traffic jams
- difficulty making left turns and lane changes
- mass transit vehicles running behind schedule
- mass transit vehicles full
- waiting through multiple cycles of traffic signals

Moreover, these effects not only happen at the locations and times where the demand/capacity ratio is high, but they also spill back into upstream traffic and later times.

2. PARAMETER SETTINGS

There are many parameters controlling the execution of the Traffic Microsimulator, but none are specific to a particular city. The parameters important for modeling traffic flow have been set based on calibration runs on special-purpose networks. These parameters control lane-changing and other aspects of the basic vehicle-following driving model. They were not tuned to Portland traffic, but to the Highway Capacity Manual. As noted in Volume Four (*General Results (not validation)*), Chapter Four (*Microsimulation*) of the Portland Study Report series, some parameters and network tables were adjusted to address issues that arose during the feedback process.

Most of the parameters specific to the Portland Case Study were thus input/output settings. For a more detailed description of the meanings of these parameters, see the Output subsystem documentation. Input parameters specify the location of the network tables, plan files, and vehicle files. Output parameters specify what to include in snapshot, summary, and event output. Some of these, such as snapshot data, are optional; others must be set to ensure that information required in the feedback processes is collected.

Among the feedback data are summary density data for every link in the network, specified by the following configuration file keys:

```

OUT_SUMMARY_NAME_8          summary.dens
OUT_SUMMARY_TYPE_8          TIME; DENSITY
OUT_SUMMARY_BEGIN_TIME_8    0
OUT_SUMMARY_END_TIME_8      86400
OUT_SUMMARY_TIME_STEP_8     900
OUT_SUMMARY_SAMPLE_TIME_8   60
OUT_SUMMARY_BOX_LENGTH_8    97.5
OUT_SUMMARY_FILTER_8        COUNT>0

```

This specification directs the Traffic Microsimulator to collect data on the density and travel times of traffic every 60 seconds, and to print out a summary of that data once every 15 minutes.

Only those links for which at least one vehicle left the link during the 15 minutes will be included in the output. The number of vehicles leaving a link is accumulated over the 60-second sampling period, so no vehicles are missed. The data from this output file will be used directly by the Route Planner to determine link travel times.

There is no way to distinguish between links with no vehicles present and completely jammed links in the data above, so we add another file with the following specifications:

```

OUT_SUMMARY_NAME_11         summary.noflow.dens
OUT_SUMMARY_TYPE_11         TIME; DENSITY
OUT_SUMMARY_BEGIN_TIME_11   0
OUT_SUMMARY_END_TIME_11     86400
OUT_SUMMARY_TIME_STEP_11    900
OUT_SUMMARY_SAMPLE_TIME_11  60
OUT_SUMMARY_BOX_LENGTH_11   97.5
OUT_SUMMARY_FILTER_11       COUNT==0; VCOUNT>0

```

These are links that had vehicles present, but did not have vehicles leaving. The travel time for these links will be reported as 0, but we can adjust that to any desired value before feeding it back to the Route Planner.

We also collect data for comparison with traffic counts along cutlines supplied by Portland. The output specifications are:

```

OUT_SUMMARY_NAME_9          cutline.auto
OUT_SUMMARY_TYPE_9          TIME
OUT_SUMMARY_BEGIN_TIME_9    0
OUT_SUMMARY_END_TIME_9      86400
OUT_SUMMARY_TIME_STEP_9     900

```


Among the optional data we track are reports of anomalies from travelers. We lump these into two files: one for travelers who become off-plan because of unanticipated congestion; another for most other anomalies. Other anomalies include events such as not being able to board a transit vehicle because it is full. A complete table is given in the documentation. We have not included anomaly 0, which is generated each time a vehicle cannot make a desired lane change because of congestion.

```

OUT_EVENT_NAME_4          anomaly.offplan
OUT_EVENT_TYPE_4         TRAVELER
OUT_EVENT_SUPPRESS_4     ACCELS;ROUTE;SIGNALS;STOPS;TURN;USER;VEHTYPE;VSTYPE;YIELDS
OUT_EVENT_FILTER_4      STATUS&8388608; ANOMALY==1
OUT_EVENT_BEGIN_TIME_4   0
OUT_EVENT_END_TIME_4     86402

OUT_EVENT_NAME_5          anomaly.other
OUT_EVENT_TYPE_5         TRAVELER
OUT_EVENT_SUPPRESS_5     ACCELS;LEG;ROUTE;SIGNALS;STOPS;TRIP;TURN;USER;VEHTYPE;VSTYPE;YIELDS
OUT_EVENT_FILTER_5      ANOMALY>2; ANOMALY<8
OUT_EVENT_BEGIN_TIME_5   0
OUT_EVENT_END_TIME_5     86402

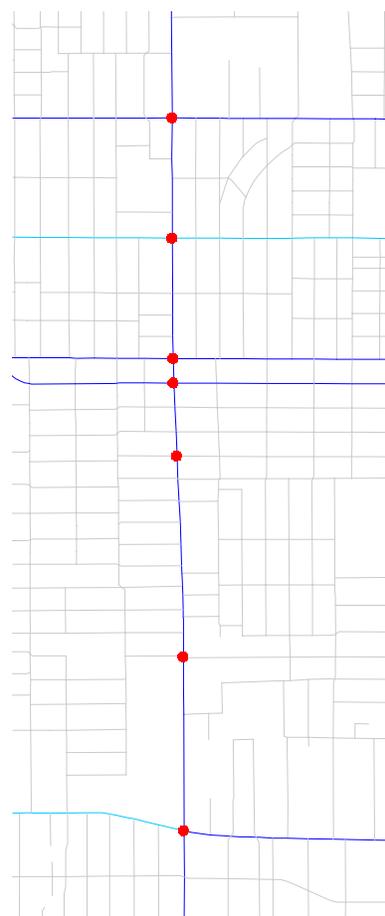
```

3. SIGNALIZATION IN THE PORTLAND STUDY

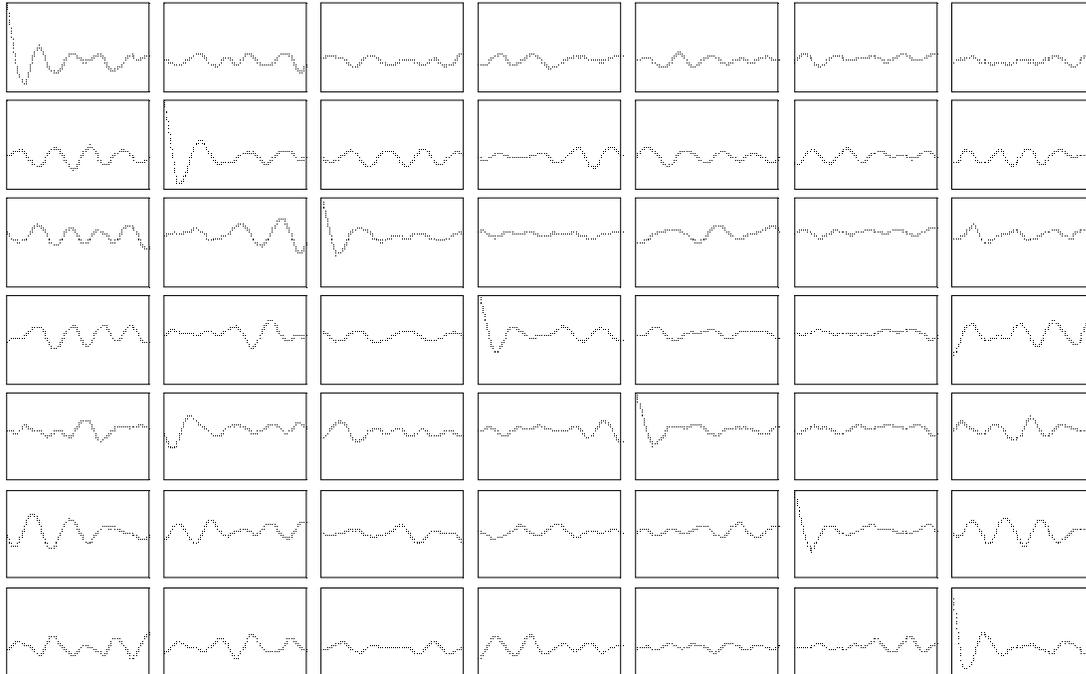
For the Portland, Oregon, case study we used the TRANSIMS traffic control generation program¹ to automatically generate 2086 actuated signals with the parameters very close to the standard set discussed above. We decided not to represent actual signals in the field because of the difficulty of obtaining and processing information describing how the signal controllers in the field had been set.

In order to take advantage of our understanding that well-chosen green times are necessary for coordinated signal operation, we use a signal-generation algorithm where the minimum green length for a phase is based on a weighted sum of the number of incoming lanes participating in the phases, but it is never less than 25 seconds. (A permanent lane has a weight of 1, and a pocket lane has a weight of 1/3.) The total cycle length of at least 60 seconds is apportioned to the phases with constant yellow and red clearance intervals, and minimum green intervals based upon the ratio of the number of incoming lanes in this phase to the total number of lanes. The green extension fraction is set to 60%. The complete set of TRANSIMS configuration settings used to generate the signals is given in Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Eleven (*MS-7*) of the Portland Study Report series.

We have examined the output of the traffic microsimulation to verify that the set of signal parameters we chose supports adequate throughput of vehicles along key corridors in the network. For example, we studied in detail the corridor containing seven signals between 82nd Avenue between Glissan Street and Division Street (illustrated at the right) between 7:00 a.m. and 7:30 a.m. The plots below show the coordinated operation of signals along this corridor: the rows correspond to the states of the seven signals (ordered from north to south) at a given time, and the columns correspond to their states at a later time represented by the distance along the horizontal axis (0 to 300 seconds); the vertical axis represents the fraction of the time that the later signal is green for southbound traffic given that the earlier signal is green for southbound traffic. As for the five-signal test case discussed above, the coordinated behavior also emerges along the 82nd-Street corridor, but this time in the context of a whole-city simulation. The average travel time for travelers moving south in this corridor is 252 seconds; the average number of seconds stopped in traffic is 66 seconds



¹ C.L. Barrett, R.J. Beckman, K.P. Berkbigler, K.R. Bisset, B.W. Bush, K. Campbell, S. Eubank, K.M. Henson, J.M. Hurford, D.A. Kubicek, M.V. Marathe, P.R. Romero, J.P. Smith, L.L. Smith, P.L. Speckman, P.E. Stretz, G.L. Thayer, E. Van Eeckhout, M.D. Williams, *TRANSIMS: Transportation ANalysis SIMulation System, Volume 3: Modules*, Los Alamos National Laboratory Report LA-UR-00-1725, 15 June 2001.



3.1 Prospects

We have demonstrated that a single control algorithm with a fixed set of parameters allows for high-performance actuated-signal intersections that are very responsive to varying traffic demand. These can be used in traffic simulations where data on actual controls cannot be easily obtained. Coordination between signals emerges where traffic conditions permit. The signals also perform adequately in real city-wide traffic simulations.

It is possible to continue and extend these calibration studies in several areas: refining the heuristics for choosing parameters, improving our parameter optimization methodology, studying behavior at a variety of intersection types, using larger networks, and looking for the natural emergence of coordination over wide areas. One could also pursue rule-based and pattern-recognition techniques for the automatic generation of controls on traffic networks. The implementation of actuated signals could be enhanced with more complex ring structures, algorithms for specific controllers, coordination of signals (i.e., wide-area control), and ITS technologies.

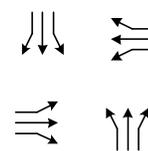
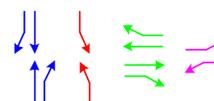
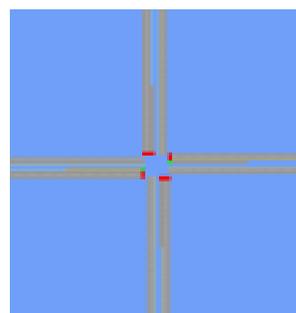
4. GENERIC SIGNAL TUTORIAL

This tutorial outlines how to create generic signals for calibration and testing, and how to run the Traffic Microsimulator to measure their performance. The files for the tutorial reside in the directory *\$TRANSIMS_HOME/data/gensig*.

4.1 Generic Signal Builder

The generic signal builder is a software application for constructing an intersection and vehicular demand for testing the performance of an actuated signal in the TRANSIMS Traffic Microsimulator. It generates the following:

- One intersection with an actuated signal and a specified timing plan.
- Four links in the cardinal directions with a specified length, number of lanes, and speed limit in the north-south and east-west directions.
- Four pocket lanes (one per link) of a specified length for left turns.
- Eight vehicle detectors (two per phase) of a specified length and placement.
- Four phases at the signal: “north-south through,” “north-south left,” “east-west through,” and “east-west left.”
- Plans for vehicles with specified headways for the twelve possible movements at the intersection: “southbound left,” “southbound through,” “southbound right,” “westbound left,” “westbound through,” “westbound right,” “northbound left,” “northbound through,” “northbound right,” “eastbound left,” “eastbound through,” and “eastbound right.”



The TRANSIMS configuration file keys controlling the details of the intersection and demand are given below. The values in the table reflect those used in this tutorial.

Key	Value in Example	Description
CALIB_PLUS_TIME_LIMIT	3600	The number of seconds for which to generate plans.
CALIB_PLUS_SPACING_NORTH_LEFT	0	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the north.
CALIB_PLUS_SPACING_NORTH_THRU	3	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the north.
CALIB_PLUS_SPACING_NORTH_RIGHT	0	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the north.
CALIB_PLUS_SPACING_EAST_LEFT	18	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the east.
CALIB_PLUS_SPACING_EAST_THRU	4	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the east.
CALIB_PLUS_SPACING_EAST_RIGHT	18	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the east.
CALIB_PLUS_SPACING_SOUTH_LEFT	24	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the south.
CALIB_PLUS_SPACING_SOUTH_THRU	7	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the south.
CALIB_PLUS_SPACING_SOUTH_RIGHT	21	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the south.
CALIB_PLUS_SPACING_WEST_LEFT	30	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the west.
CALIB_PLUS_SPACING_WEST_THRU	24	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the west.
CALIB_PLUS_SPACING_WEST_RIGHT	27	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the west.
CALIB_PLUS_LANES_NS	2	The number of lanes in the north-south direction.
CALIB_PLUS_LANES_EW	2	The number of lanes in the east-west direction.
CALIB_PLUS_SPEED_NS	<default>	The speed limit [meters per second] in the north-south direction. The default is five cells per second.
CALIB_PLUS_SPEED_EW	<default>	The speed limit [meters per second] in the east-west direction. The default is five cells per second.
CALIB_PLUS_GREEN_THRU_NS	20	The number of seconds for greens of the "north-south through" phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_LEFT_NS	8	The number of seconds for greens of the "north-south left" phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_THRU_EW	20	The number of seconds for greens of the "east-west through" phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_LEFT_EW	8	The number of seconds for greens of the "east-west left" phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_EXTENSION	0.60	The green extension in terms of the fraction of the initial green. The default is 0.50.
CALIB_PLUS_DETECTOR_LENGTH	37.5	The length [meters] of the detectors. The default is six cells.
CALIB_PLUS_DETECTOR_OFFSET	<default>	The offset [meters] of the detectors, measured from the point of the detector closest to the node to the node. The default is the intersection setback.
CALIB_PLUS_LINK_LENGTH	<default>	The length [meters] of the links. The default is thirty cells plus the intersection setback.
CALIB_PLUS_POCKET_LENGTH	<default>	The length [meters] of pocket lanes. The default is eight cells.
CALIB_PLUS_PARKING_OFFSET	<default>	The offset [meters] of the parking from the edges of the network.
CALIB_PLUS_SETBACK	<default>	The setback [meters] at the intersection. This defaults to an optimal value for the TRANSIMS Visualizer.

The generic signal builder also uses the additional TRANSIMS configuration file keys in the table below—not all of these are required, however.

Key	Required?
NET_DIRECTORY	yes
NET_NODE_TABLE	yes
NET_LINK_TABLE	yes
NET_POCKET_LANE_TABLE	yes
NET_PARKING_TABLE	yes
NET_LANE_CONNECTIVITY_TABLE	yes
NET_UNSIGNALIZED_NODE_TABLE	yes
NET_SIGNALIZED_NODE_TABLE	yes
NET_PHASING_PLAN_TABLE	yes
NET_TIMING_PLAN_TABLE	yes
NET_ACTIVITY_LOCATION_TABLE	yes
NET_PROCESS_LINK_TABLE	yes
NET_STUDY_AREA_LINKS_TABLE	yes
NET_BARRIER_TABLE	yes

Key	Required?
NET_DETECTOR_TABLE	yes
NET_LANE_USE_TABLE	yes
NET_SIGNAL_COORDINATOR_TABLE	yes
NET_SPEED_TABLE	yes
NET_TRANSIT_STOP_TABLE	yes
NET_TURN_PROHIBITION_TABLE	yes
NET_LANE_WIDTH	no
NET_LINK_MEDIAN_HALFWIDTH	no
VEHICLE_FILE	yes
VEHICLE_PROTOTYPE_FILE	yes
PLAN_FILE	yes
OUT_SNAPSHOT_NODES_1	yes
OUT_SNAPSHOT_LINKS_1	yes
CA_CELL_LENGTH	no

The generic signal builder is invoked with a single argument:

```
BuildTestSignal <config-file>
```

where <config-file> is the name of the configuration file containing the keys listed in the two tables above. The directory *\$TRANSIMS_HOME/data/gensig* contains the configuration file *signal.cfg* with the keys used in this tutorial. Running *BuildTestSignal* with this configuration file results in the network files stored in the *network* subdirectory and in the vehicle and plan files stored in the *data* subdirectory.

4.2 Assessing Generic Signal Response

In order to assess the response of the generic signal created by the builder, we run the Traffic Microsimulator and analyze the output. The script *signal.csh* in the directory *\$TRANSIMS_HOME/data/gensig/scripts* runs the does this. After running the Traffic Microsimulator, the *output* subdirectory contains several files of interest:

- *snapshot.veh*: the vehicle snapshot data for the travelers—suitable for viewing in the TRANSIMS Output Visualizer
- *snapshot.sig*: the signal snapshot data for the actuated signal—suitable for viewing in the TRANSIMS Output Visualizer
- *times.trv*: the travel time and inconvenience measure statistics for the travelers—suitable for analysis with the *scripts/counts.awk* script
- *lost.trv*: traveler event data for any travelers that became lost in the simulation—there should be few or none

The script will also produce console output summarizing the response of the signal. Four performance measures are tabulated for each movement at the intersection:

- *flow [veh/hr]*: the flow of vehicles making the movement
- *travel time [s/veh]*: the observed mean and standard deviation for the travel times measured from when a vehicle leaves its parking location on the incoming link to when it enters its parking location on the outgoing link
- *time stopped [s/veh]*: the observed mean and standard deviation for the number of seconds that a vehicle is stopped while it waits to pass through the intersection
- *accelerations from stop [#veh]*: the observed mean and standard deviation for the number of times a vehicle has to accelerate after being stopped on the roadway

For example, in this tutorial the summary should look similar to the following:

```
Movement:                southbound, left
  Flow [veh/hr]:          0

Movement:                southbound, thru
  Flow [veh/hr]:          1177
  Travel Time [s/veh]:    35.3696 (mean), 15.3534 (s.d.)
  Time stopped [s/veh]:   3.19031 (mean), 9.24733 (s.d.)
  Accels. from stop [#veh]: 2.13594 (mean), 0.859248 (s.d.)

Movement:                southbound, right
  Flow [veh/hr]:          0

Movement:                westbound, left
  Flow [veh/hr]:          197
  Travel Time [s/veh]:    86.4162 (mean), 43.3299 (s.d.)
  Time stopped [s/veh]:   11.1015 (mean), 25.0976 (s.d.)
  Accels. from stop [#veh]: 3.30457 (mean), 1.48758 (s.d.)

Movement:                westbound, thru
  Flow [veh/hr]:          894
  Travel Time [s/veh]:    52.4586 (mean), 26.8092 (s.d.)
  Time stopped [s/veh]:   2.84228 (mean), 9.19959 (s.d.)
  Accels. from stop [#veh]: 2.9396 (mean), 1.5013 (s.d.)

Movement:                westbound, right
  Flow [veh/hr]:          199
  Travel Time [s/veh]:    54.5578 (mean), 28.1368 (s.d.)
  Time stopped [s/veh]:   5.09045 (mean), 13.0257 (s.d.)
  Accels. from stop [#veh]: 2.59799 (mean), 1.32543 (s.d.)

Movement:                northbound, left
  Flow [veh/hr]:          139
  Travel Time [s/veh]:    150.367 (mean), 87.6663 (s.d.)
  Time stopped [s/veh]:   17.3813 (mean), 43.9112 (s.d.)
  Accels. from stop [#veh]: 3.54676 (mean), 2.14078 (s.d.)
```

```

Movement:                northbound, thru
  Flow [veh/hr]:          499
  Travel Time [s/veh]:    67.6212 (mean), 55.6657 (s.d.)
  Time stopped [s/veh]:   4.11222 (mean), 10.2163 (s.d.)
  Accels. from stop [#veh]: 3.11824 (mean), 2.12748 (s.d.)

Movement:                northbound, right
  Flow [veh/hr]:          167
  Travel Time [s/veh]:    66.9701 (mean), 57.4042 (s.d.)
  Time stopped [s/veh]:   3.07784 (mean), 8.74332 (s.d.)
  Accels. from stop [#veh]: 2.8024 (mean), 1.66552 (s.d.)

Movement:                eastbound, left
  Flow [veh/hr]:          119
  Travel Time [s/veh]:    61.8739 (mean), 36.7648 (s.d.)
  Time stopped [s/veh]:   18.1008 (mean), 30.0298 (s.d.)
  Accels. from stop [#veh]: 1.72269 (mean), 0.649959 (s.d.)

Movement:                eastbound, thru
  Flow [veh/hr]:          149
  Travel Time [s/veh]:    33.2886 (mean), 19.4514 (s.d.)
  Time stopped [s/veh]:   10.0067 (mean), 15.5748 (s.d.)
  Accels. from stop [#veh]: 1.39597 (mean), 0.555304 (s.d.)

Movement:                eastbound, right
  Flow [veh/hr]:          133
  Travel Time [s/veh]:    33.0752 (mean), 19.1769 (s.d.)
  Time stopped [s/veh]:   10.1955 (mean), 15.1164 (s.d.)
  Accels. from stop [#veh]: 1.43609 (mean), 0.655436 (s.d.)

```

For reference, the demand in this example is given below.

<i>Movement</i>	<i>Demand [veh/hr]</i>
Southbound, left	0
Southbound, through	1200
Southbound, right	0
Westbound, left	200
Westbound, through	900
Westbound, right	200

<i>Movement</i>	<i>Demand [veh/hr]</i>
Northbound, left	150
Northbound, through	514
Northbound, right	171
Eastbound, left	120
Eastbound, through	150
Eastbound, right	133

We can see that nearly all of the vehicles make it through the intersection with the actuated signal.

By editing the CALIB_PLUS_SPACING parameters in the configuration file and then rerunning *BuildTestSignal*, one can evaluate additional demands for the signal under consideration. One can change the actuation algorithm's parameters by editing its configuration parameters:

- *NET_DETECTOR_PRESENCE_SAMPLE_TIME*
- *NET_ACTUATED_ALGORITHM_B_BETA*
- *NET_ACTUATED_ALGORITHM_B_DENSITY_CONST*
- *NET_ACTUATED_ALGORITHM_B_FLOW_CONST*

The signal timing and detector layout may be altered with other CALIB_PLUS configuration file keys.

Part 3

General Results

Chapter 1

Population and Activities

1. GENERAL RESULTS

1.1 Agreement with Forecast

The synthetic population covers 1107 block groups in the greater Portland area. There are a total of 636,389 households, compared to 636,533 households in Portland Metro’s forecast (a difference of 144 households). Over 60% of the block groups have exactly the population specified in the forecast. The marginal counts of households for the size, age, and income demographics match very closely: 10.9% of the counts have a p -level below 10%, 0.99% of the counts have a p -level below 1%, and 0.21% of the counts have a p -level below 0.1% for example, if we check the synthetic population with a chi-squared test. There are no differences in the structural zeros for the two sets of marginal distributions.

For a more stringent check of the agreement of the synthetic population with the forecast, we have examined the odds ratio,

$$\phi = \frac{P_{1,1} \cdot P_{2,2}}{P_{1,2} \cdot P_{2,1}},$$

for randomly-chosen cells in the multiway table for persons, age, and income. (The quantities $P_{1,1}$, $P_{2,2}$, $P_{1,2}$, and $P_{2,1}$ represent the counts in related cells of the multiway table.) Since the iterative proportional fitting procedure preserves the correlation structure (and hence the odds ratio) of the data it fits, there should be a strong correlation between the odds ratios measured in the forecast joint distribution computed by Portland Metro and the synthetic population generated in TRANSIMS. Note that the forecast joint distribution is not used by the Population Synthesizer, so this check constitutes an independent test of the synthesis procedure. Fig. 1 shows a comparison of the cells of the multiway table, while Fig. 2 shows a comparison of the odds ratios—in each case we have good agreement between the forecast and the synthesis.

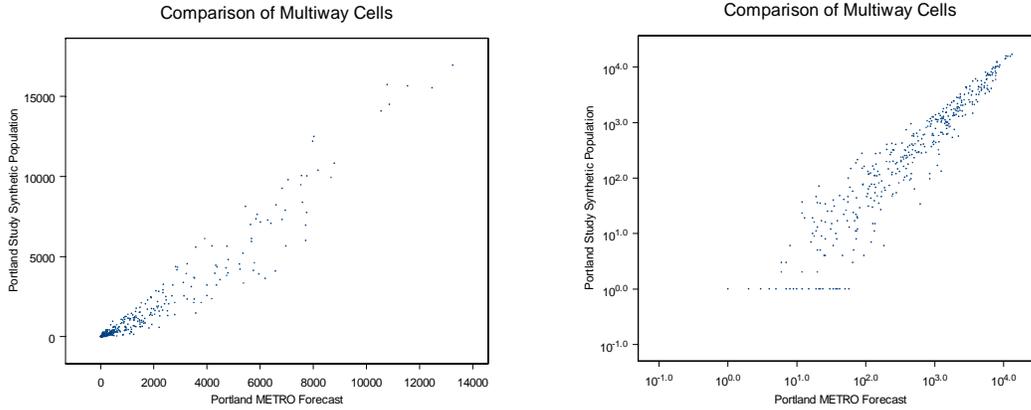


Fig. 1. Comparison of counts in the 441-cell persons \times age \times income multiway table representing the correlation structures of the Portland Metro forecast and the Portland study synthetic population: linear scale (left) and logarithmic scale (right). The strong correlation indicates good agreement.

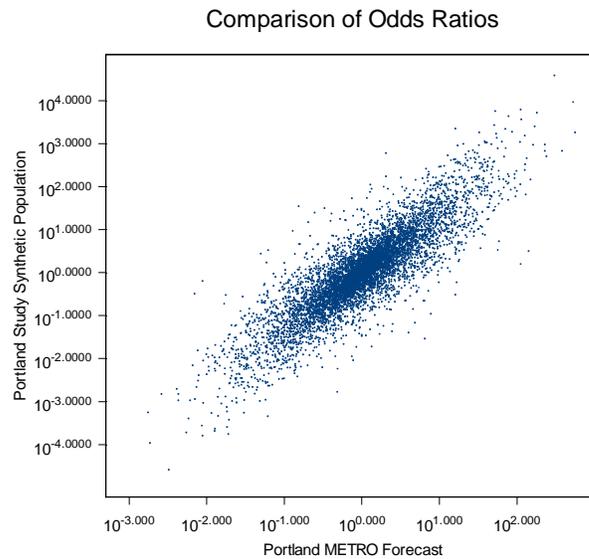


Fig. 2. Comparison of 21,898 randomly-chosen odds ratios for the 441-cell persons \times age \times income multiway table representing the correlation structures of the Portland Metro forecast and the Portland study synthetic population. The strong ellipsoidal shape indicates good agreement.

1.2 Example Block Group

We choose block group 31200.1 (shown in Fig. 3) to illustrate the results of synthesizing and locating a population; Table 1 lists the forecast marginal demographic data provided by Portland Metro. Fig. 4 shows how the correlation structure of demographic variables is preserved after iterative proportional fitting, while Fig. 5 demonstrates that the marginal distributions of the demographic variables closely match the forecast.

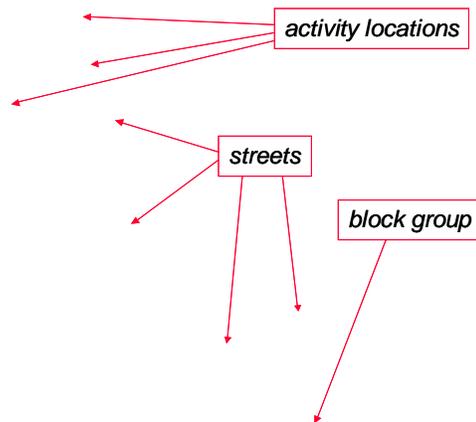


Fig. 3. Geographic layout of block group 31200.1 in Portland, Oregon.

Table 1. Forecast (1996) marginal demographics for block group 31200.1 in Portland, Oregon.

Size	Households	Age of Head	Households	Income	Households
1	84	β 24	24	β 4999	0
2	42	25–34	42	5000–9999	47
3	0	35–44	35	10,000–14,999	8
4	6	45–54	28	15,000–24,999	19
5	0	55–64	0	25,000–34,999	37
6	0	65–74	2	35,000–49,999	0
ρ 7	0	ρ 75	1	50,000–74,999	21
Total	132	Total	132	75,000–99,999	0
				ρ 100,000	0
					132

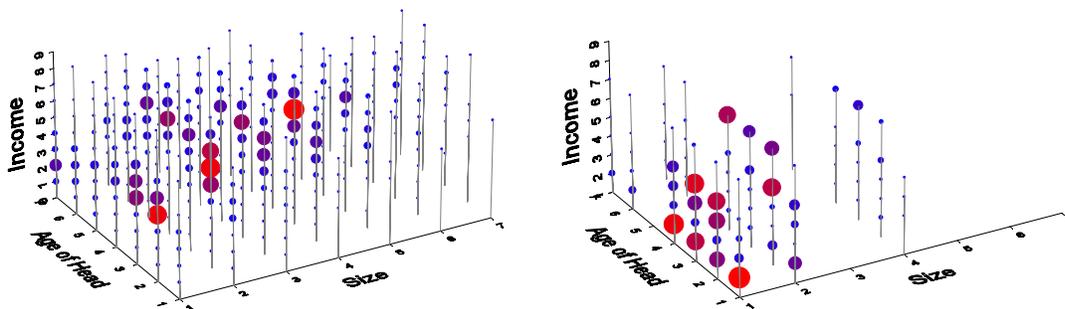


Fig. 4. Comparison of multiway tables for the whole study area (left) and for block group 31200.1 (right) in Portland, Oregon.

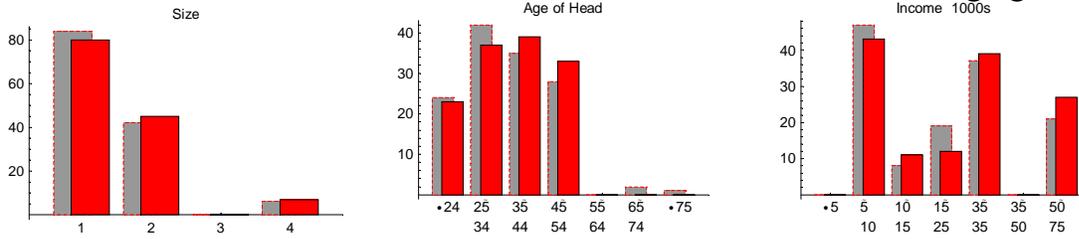


Fig. 5. Comparison between forecast marginal distributions (gray) and counts of households in the synthetic population (red) for block group 31200.1 in Portland, Oregon.

In Table 2 and Table 3, we list the demographic variables generated for an example household from block group 31200.1. Using the population synthesizer, it is possible to generate populations with any of 65 household variables and any of 74 person variables. The variables listed in the tables are those used in the Portland study. Finally, we show in Fig. 6 that the count of households located along streets in block group 31200.1 closely matches the residence area obtained from tax lot data.

Table 2. Selected person demographics for a synthetic household in block group 31200.1 in Portland, Oregon.

ID	111733
Size	4
Vehicles	3
Activity Location	23101
PUMS Record	44789
Anyone under 18	Yes
Workers in 1989	3+
Total Income	\$64,000
Tenure	Owned with mortgage or loan
Value	\$90,000 - \$99,999

Table 3. Selected person demographics for a synthetic household in block group 31200.1 in Portland, Oregon.

ID	255552	255553	255554	255555
Age	42	42	19	7
Relationship	Householder	Husband/wife	Son/daughter	Son/daughter
Sex	Male	Female	Female	Female
Worked in 1989	Yes	Yes	Yes	No (under 18)
Educational Attainment	Some college, but no degree	High school graduate, diploma or GED	Some college, but no degree	1 st , 2 nd , 3 rd , or 4 th grade
Industry	Electrical Machinery, Equipment, and Supplies, N.E.C	Not Specified Retail Trade	Offices and Clinics of Chiropractors	
Occupation	Managers and Administrators, N.E.C	Sales Workers, Other Commodities	Managers, Medicine and Health	
Total Income	\$45,000	\$13,000	\$6000	
Hours Worked	40	40	15	
Lived Here in 1985	No	No	No	(under 5)
Means of transportation to work	Car, truck, or van	Car, truck, or van	Car, truck, or van	
Vehicle occupancy	1	1	1	
Time of departure for work	6:50	1:00	14:00	
Travel time to work	0:20	0:15	0:10	

Fig. 6. Residence area (gray bars) and count of synthetic households (red bars) for part of block group 31200.1 in Portland, Oregon.

2. ACTIVITIES

Activities for the Portland population were generated in parallel by splitting the population into 20 household lists. Activities for each of the 20 household lists were generated in about 40 minutes simultaneously using 20 compute nodes on the rockhopper cluster. The resulting activity files were combined to produce a single activity file that was approximately 973 megabytes in size.

Activities for itinerant travelers and freight trips were produced from trip tables. A pair of activities (origin and destination) was produced for each trip in the tables.

- 205,957 itinerant traveler trips were generated, resulting in 411,914 activities.
- 41,997 freight trips were generated resulting in 83,994 activities.

A base activity set is generated for the entire synthetic population. Trips are counted by mode from this activity. The trips are compared with calibration numbers, and the biasing scheme described in Volume 2 (*Study Setup: Parameters and Input Data*), Chapter 3 (*Activity Generation*) of the Portland Study Report series is executed. In this study, it was determined that the transit (including MAX) plus the drive trips were 25% too low in the original activity set. The biasing scheme was used to increase these trips by 25%.

Table 4 shows the number of trips by mode before (Weights=1) and after (Weights Computed), the biasing scheme, was executed. In both cases, the Magic Move mode represents interhousehold shared rides that were not simulated in this version of TRANSIMS.

Table 4. The number of trips by mode before and after biasing the selection of households with more trips. The goal of the biasing is to increase the transit (including MAX) plus the drive trips by 25%.

Mode	Weights=1 Trips	Mode	Weights Computed Trips
Walk	542,077	Walk	479,160
Drive	4,573,938	Drive	6,039,928
Transit	167,673	Transit	156,736
MAX	29,315	MAX	28,504
Bike	68,385	Bike	47,847
Magic Move	292,464	Magic Move	306,119

3. SUMMARY

In conclusion, we have generated a statistically realistic synthetic population for use in the Portland study. The synthetic population preserves the key features of the actual population in that it matches its correlation structure, the joint distributions of demographic variables, the spatial distribution of households, and the number of vehicles for households. We have tested to output data from the Population Synthesizer to verify that it performs the necessary calculations correctly and that the results statistically “match” the required constraints. The synthesized population possess a rich and detailed selection of demographic information for individual households and persons.

We also note that the Population Synthesizer has practical input data requirements. It uses readily-available census data (STF-3A, PUMS, TIGER) along with data typically available at metropolitan planning organizations (block-group-level marginal forecasts of population demographics, tax lot and zoning information, and motor vehicle records).

Chapter 2

Routes

1. NETWORK

For efficiency reasons, the TRANSIMS network (see Volume Two (*Study Setup: Parameters and Input Data*), Chapter One (*Network*) of the Portland Study Report series) is converted to an internal Route Planner network. The TRANSIMS documentation contains details about this transformation. There are 475,246 nodes and 650,994 links in the TRANSIMS network; and 506,138 nodes, 1,104,177 links, and 3,287,972 edges in the internal Route Planner network. The internal edges are the network over which the actual routing takes place. Table 1 breaks down these numbers by layer for the internal Route Planner network.

Table 1. Internal Route Planner network by layer.

Layer	Nodes	Links	Edges
Street	100511	244748	249222
Parking	121503		722907
Activity	243423	829180	2285594
Bus	9771		
Light Rail	56		
Route	30874	30249	30249
<i>Total</i>	<i>506138</i>	<i>1104177</i>	<i>3287972</i>

2. PLAN FILES

There are a total of seven individual plan sets that make up a particular route set: four Itinerants (AM, Midday, PM, Rest), Trucks, Transit, and Population. Only the population, which may be split into multiple files, is generated more than once. The Transit plan set is not generated by the Route Planner at all, but created from the transit routes. The sizes of each plan set in RS-7 and RS-19 are shown in Table 2 and **Table 3**.

Table 2. Plan set sizes – RS-7.

	Bytes	GB	Trips
AM	34767533	.03	32716
Mid	123345438	.11	114723
PM	33427866	.03	33531
Rest	26524411	.02	24987
Trucks	53326331	.04	41993
Transit	9132414	.01	9429
Population	3600500835	3.35	
<i>Total</i>	<i>3881024828</i>	<i>3.61</i>	

Table 3. Plan set sizes – RS-19.

	Bytes	GB	Trips
AM	34767533	.03	32716
Mid	123345438	.11	114723
PM	33427866	.03	33531
Rest	26524411	.02	24987
Trucks	53326331	.04	41993
Transit	9132414	.01	9429
Population	4536471783	4.22	
<i>Total</i>	<i>4816995776</i>	<i>4.48</i>	

3. ANOMALIES

Anomalies occur when a problem is encountered while generating a route for a traveler. An anomaly can either be a warning or an error. A warning indicates that there may be a problem with a route, but the route is still written to the plan file, and the rest of the traveler's trips continue to be planned. The most common warnings are trip lengths that cause the time for an activity to fall outside of the range specified in the activity file. An error indicates that a route could not be created. Any further trips for that traveler are not planned. The most common error is that a path through the network either cannot be found or takes too long.

Table 4 shows the anomalies reported for route sets RS-7 and RS-19. Table 5 and

Table 6 break down the errors by travel mode for RS-7 and RS-19. The percentage error is the percentage of errors for trips of that mode, based on the total trips by mode from the activity file.

Table 4. Anomaly summary.

	AM	Mid	PM	Rest	Trucks	RS-7	RS-19
Total Problems	32717	114724	33532	24989	42009	2978799	496145
Errors	0	0	0	0	0	79919	13175
No Path Exits	0	0	0	0	0	3179	8760
Trip Time	0	0	0	0	0	3559	1016
Leg Time	0	0	0	0	0	56784	0
Too Many Nodes	0	0	0	0	0	4342	0
Pass Very Late	0	0	0	0	0	12055	3399
Warnings	32717	114724	33532	24989	42009	2898880	479923
Invalid Time Start	32716	114723	33531	24987	41997	1837822	240839
Stop	0	0	0	0	0	82196	27789
Duration	0	0	0	0	0	321084	57394
Driver Late	0	0	0	0	0	139240	54833
Pass Late	0	0	0	0	0	512753	98038
Parking Location	1	1	1	2	12	5785	1030

Table 5. RS-7 errors by mode.

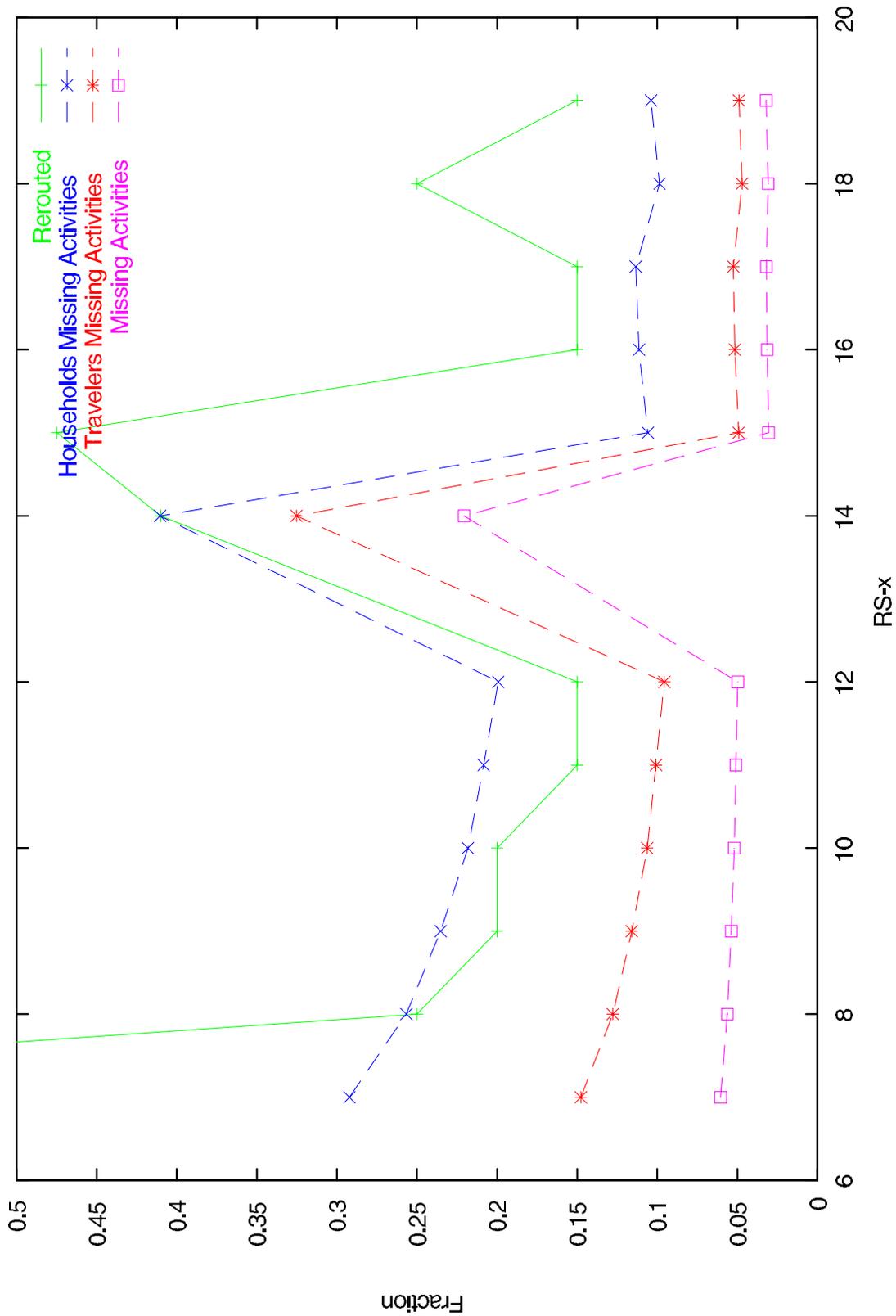
Mode	cw	w	wiw	wlw	wtw	wkw	wKw	Total
Path	1411	1363	56	84	265	0	0	3179
Trip	0	3478	4	1	76	0	0	3559
Leg	0	32401	194	6926	17263	0	0	56784
Max Node	0	0	0	0	4342	0	0	4342
V Late	12055	0	0	0	0	0	0	12055
Total	13466	37242	254	7011	21946	0	0	79919
Trips	6079443	449174	48640	28200	157042	239621	183114	7185234
% Errors	0.22	8.29	0.52	24.86	13.97	0.00	0.00	1.11

Table 6. RS-19 errors by mode.

Mode	cw	w	wiw	wlw	wtw	wkw	wKw	Total
Path	1274	1296	27	735	3885	0	0	7217
Trip	0	1001	5	0	9	0	0	1015
Leg	0	0	0	0	0	0	0	0
Too Late	0	0	0	0	0	0	0	0
Max Node	0	0	0	0	0	0	0	0
V Late	3399	0	0	0	0	0	0	3399
Total	4673	2297	32	735	3894	0	0	11631
Trips by Mode	6079443	449174	48640	28200	157042	239621	183114	7185234
% Errors	0.08	0.51	0.07	2.61	2.48	0.00	0.00	0.16

The following figure shows how the number of unplanned activities changed in the population routes from RS-7 to RS-19.

- *Households Missing Activities* is the fraction of all households in the population in which one or more activities were not planned.
- *Travelers Missing Activities* is the fraction of all travelers in the population in which one or more activities were not planned.
- *Missing Activities* is the fraction of all population activities that were not planned.
- *Rerouted* is the fraction of households that were rerouted on each iteration.



4. EXECUTION TIME

The execution time for creating the pieces of RS-7 is shown in Table 7. Many of the plan sets were created using multiple processors. The *CPU Seconds* and *CPU Hours* rows refer to the time it would have taken to run on one dual-CPU node. *Elapsed Seconds* and *Elapsed Hours* is the actual time taken using the number of Nodes and CPUs shown.

Table 7. Summary of execution time.

	AM	Mid	PM	Rest	Truck	Population	Total
CPU Seconds	48126.6	596130.5	52161.8	39916.6	153847.4	3707824	4598006.9
CPU Hours	13.4	165.6	14.5	11.1	42.7	1030	1277
Elapsed Seconds	24063.3	74516	26080.9	19958.3	76923.7	30899	
Elapsed Hours	6.7	20.7	7.2	5.5	21.4	8.6	
Nodes	1	4	1	1	1	60	68
CPUs	2	8	2	2	2	120	136

Chapter 3

Microsimulation

1. CONFIGURATION FILE KEYS

The following is a typical set of values for important Traffic Microsimulator configuration file keys. See Volume Eight (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Eleven (*MS-7*) of the Portland Study Reports for a complete configuration file.

CA_SIM_STEPS	86400
CA_OFF_PLAN_EXIT_TIME	10
CA_MAX_WAITING_SECONDS	86401
CA_LOOK_AHEAD_CELLS	35
CA_HELP_LOST_TRAVELERS	1
CA_INTERSECTION_CAPACITY	1
LOG_CONTROL	1
LOG_ROUTING	1
LOG_TIMESTEP	1
LOG_TIMING	1
PLAN_FILE	\$TRANSIMS_ROOT/plans/RS13/RS13
VEHICLE_FILE	\$TRANSIMS_ROOT/vehicle/vehicles.pop
VEHICLE_PROTOTYPE_FILE	\$TRANSIMS_ROOT/data/allstr.prototypes.short
PAR_SLAVES	47
PAR_MIN_CELLS_TO_SPLIT	10
PAR_USE_METIS_PARTITION	1
PAR_USE_OB_PARTITION	0
PAR_PARTITION_FILE	\$TRANSIMS_ROOT/data/partition.47

1.1 Notes on Configuration File Key Values

The Traffic Microsimulator crashes if any one of its output files grows larger than 2 Gbytes. We break up the snapshot data into several time periods to avoid this. Especially later in the day and

to clear jams very quickly. The configuration file key `CA_MAX_WAITING_SECONDS` must, in any case, be longer than the longest red light phase, so at most only one vehicle can be removed every few minutes from a jammed link in this fashion.

The configuration file key `CA_OFF_PLAN_EXIT_TIME` should be set low so that vehicles that do become lost for any reason do not contribute to jams; they exit the simulation at the next parking lot they pass. They will continue with the rest of their tour, however, as if they had reached their destination, because the `CA_HELP_LOST_TRAVELERS` flag is set.

It became clear in early runs of the Traffic Microsimulator that vehicles that became off-plan while on a freeway had little chance of exiting the freeway to find a parking lot. We, therefore, created an alternative table of parking lots for use only by the Traffic Microsimulator that provides a parking lot for every freeway or expressway link. Because the Route Planner never sees these extra lots, they are not included in any driver's plan. Hence, they will be used only by drivers who are off-plan.

The vehicle file specifies that all vehicles have a length of one cell. We determined that the effort required to remove all short links was not worth the slight gain in verisimilitude of longer vehicles. Heavy-duty vehicles have appropriately modified maximum speeds and powers.

2. BUG FIXES

We have fixed a few errors in the Traffic Microsimulator that reduced capacity:

- Vehicles executing a U-turn on a link that allowed U-turns on both ends became stuck on that link.
- Vehicles entering a link in a two-lane merge section sometimes became stuck hopping between lanes.
- Vehicles occasionally became stuck in intersections. This was related to the U-turn problem.

3. CHANGES IN LOGIC

We made a few changes in driving logic based on observed unrealistic behavior.

Vehicles that cannot change into their desired lane because of congestion are allowed to proceed onto any lane on their next link when they reach an intersection. If there is no connectivity from

4. COMPUTING ENVIRONMENT

All Traffic Microsimulator runs were carried out on a 128-node Linux cluster using up to 48 of the nodes. Each node contains two 500 MHz CPUs sharing 1 Gigabyte of memory. In order to reduce the memory required by the Traffic Microsimulator, we distributed the computation to 47 slaves, placing only one slave on each node. The resulting memory usage per node was roughly 200 – 400 Megabytes. This indicates that we could have used both processors on each node, but we did not typically do so. We also could have distributed work to more slaves, but the cluster hardware was less likely to fail when we minimized the number of slaves.

Total output disk space required for any run was under 5 Gigabytes. A breakdown by file and run is provided in the three tables below. The output disk space requirements were dominated by the vehicle snapshot data, summary data, end-of-trip events, and logging messages. Summary data and end-of-trip events are required for Route Planner feedback; snapshot data proved valuable for understanding the progress of the iterations; logging messages could be turned off with no ill effects.

Table 1. File sizes for those files that did not vary significantly from run to run.

File name	File size
cutline.transit.trv	46 K
cutline.truck.tim	1 – 2 M
cutline.auto.tim	1 – 2 M
run_time_monitor	4 M
summary.noflow.dens.spa	116 K

Table 2. File sizes in Kilobytes (K) or Megabytes (M) for non-snapshot Traffic Microsimulator output files that varied in size from run to run. Some of these provide gross measures of performance. For example, the number of off-plan travelers should decrease with iteration.

Iteration number	anomaly.offplan.trv (M)	anomaly.other.trv (K)	busmall.snapshot.veh (M)	endsim.trv (M)	endtrip.trv (M)	Logfile (M)	snapshot.am.int (K)	summary.dens.spa (M)	summary.dens.tim (M)	summary.noflow.dens.tim (M)
20	72	65	23	35	773	592	383	429	507	112
19R	70	3	23	39	750	618	380	417	491	113
19	68	89	23	38	743	609	396	413	489	111
18	61	99	21	44	663	644	450	387	428	137
17	38	15	2	55	529	101	720	350	352	167
16	34	3000	2	50	663	90	702	321	321	171

Table 3. File sizes in Megabytes (M) for snapshot output. The day was broken into several different time intervals to ensure that individual output files remained smaller than 2 Gigabytes. The file size is proportional to the average number of vehicles present per timestep; extremely large files indicate the presence of large traffic jams.

Iteration number	allstr.snapshot.veh (M)	snapshot.am.veh (M)	snapshot.pm.veh (M)	snapshot.12to4pm.veh (M)	snapshot.4to8pm.veh (M)	snapshot.8to9pm.veh (M)
20		154		254	478	156
19R		155		261	522	174
19		160	913			
18		195	1180			
17		301	1600			
16		348	1780			
15		424	1940			
11R		342				
7	521					