Microsimulator How-To

This document provides basic information about using the Microsimulator to simulate person and vehicle movements on a network using travel plans generated by the Router. It also discusses how to review and resolve simulation problems and how to achieve user equilibrium through multiple feedback loops.

Revision History

11/10/2006	Created by AECOM Consult, Inc.
12/1/2006	Revised by AECOM Consult, Inc. based on 12/1/06 Release
2/07/2007	Revised by AECOM Consult, Inc.
4/15/2010	Revised by RSG, Inc.

Table of Contents

1.0 Assumptions and Prerequisites

This document assumes you have installed TRANSIMS Version 4.0.3 on a Windows or Linux computer system and that you understand the basic procedures and terminology for executing TRANSIMS programs.

The TRANSIMS software and documentation can be downloaded from http://sourceforge.net/projects/transims/files/ → software. Basic TRANSIMS procedures and terminology are addressed in the Installation and Testing How-To available in the documentation set.

You should have a basic understanding of traditional transportation planning networks and equilibrium methods. You should also have successfully run previous how-to exercises through the Router and Router Feedback How-To.

The input and output data are stored in text files. You need to be able to review and edit these files using a standard text editor (e.g., vi, Pico, WordPad) or other software that can manipulate text files (e.g., Excel).

Viewing the vehicle trajectories generated by the Microsimulator through visualization software is helpful, but not necessary. The TRANSIMS Output Visualizer software can display second-by-second snapshots generated by the Microsimulator. This exercise demonstrates how to display output summaries using ArcGIS.

1.1 Download the Case Study Files

Network information from Alexandria, Virginia, is used to demonstrate the procedures for using the Microsimulator, discuss outcomes, and describe concepts. The travel plans needed by this case study are generated by the Router and Router Feedback How-To.

To download the Alexandria data to your computer or local area network, select http://sourceforge.net/projects/transims/files/ → test data → 4.0.3a Test Cases → Alexandria_4.0.3a.zip

You should create a directory with a name such as

c:\TRANSIMS\Alexandria	(Windows)
/home/TRANSIMS/Alexandria	(Linux)

and then unzip the downloaded file to that directory.

This test case assumes the travel plans generated during the Router and Router Feedback How-To are located in the plans directory.

2.0 Introduction to TRANSIMS Microsimulation

The TRANSIMS Microsimulator works in conjunction with the Router to simulate the travel conditions experienced by individual travelers. This section introduces basic Microsimulator concepts and then describes how the simulation is combined with the Router to implement a user-equilibrium assignment.

2.1 Simulating Vehicle Travel

The Microsimulator program simulates the movement and interaction of persons and vehicles in a multimodal transportation network that includes pedestrians, cars, trucks, buses, and trains. The simulation is carried out in discrete, user-specified time intervals of one second or less over the course of a day. In each time step, a vehicle in the network accelerates, decelerates, changes lanes, or stops based on the behavior of nearby vehicles and of traffic controls.

Each link in the transportation network is depicted as a grid of cells, with each pocket and through lane containing it own set of cells. The cell size is typically set equal to the total space occupied by a passenger car when stopped in a queue. Each vehicle thus occupies one or more cells depending on its length. Relatively simple rules are used to move vehicles between cells in the network. The Microsimulator generates performance statistics based on cell movements per second that replicate fundamental flow-density-speed relationships.

A vehicle's path through the network is defined by the travel plan generated by the Router. The travel plan identifies the time at which the trip begins and the sequence of links or transit routes that will be used.

The Microsimulator assigns vehicles to travel lanes and sets each vehicle's speed based on the speed limit, the speed of nearby vehicles, and on the presence of traffic controls. A vehicle changes lanes to position itself on a lane that leads to a downstream link defined in its travel plan. As a vehicle moves closer to a lane restriction, the urgency for it to move into a lane that will allow it to continue its trip increases. If congestion or network errors prevent a vehicle from changing lanes as necessary within a user-specified time period, the vehicle will be removed from the network and considered "lost." Failure to remove such vehicles from the network has been found to generate unrealistic congestion that spreads into the off-peak time periods and deteriorates the quality of the simulation.

A vehicle that is removed from the simulation typically points to a problem with the travel plan or the network that should be reviewed. It may be caused by a very short weaving section that requires two or more lane changes to accomplish the maneuver. If the geometrics are correct, all that may be needed is to re-route the lost vehicle so it selects a more realistic path. If the geometrics are not correct, the network can be corrected and the travel plans re-simulated. In either case, it is better to remove the vehicle from the simulation to avoid distorting the travel conditions experienced by other travelers. If more than about two percent of the vehicles are being lost during the simulation, additional iterations or other corrective actions should be taken to stabilize the results.

The Microsimulator adds transit vehicles that follow the transit drive path through the network for each scheduled run. The transit vehicles interact with the other vehicles on the network and can block traffic while stopped at a bus stop to serve passengers or wait for a scheduled time point. As a result, a transit vehicle can arrive early or late at transit stop. In either case the vehicle will wait at the stop while passengers board and alight. Since transit vehicles are capacity constrained, passengers who are unable to board a vehicle because it is full will wait for the next run of the specified transit route.

The Microsimulator can be applied at a regional level or at a subarea level. Simulations based on regional travel demand forecasting networks behave realistically in most situations. Complex network geometrics will need to be reviewed and adjusted to simulate the demand properly.

Simulations based on subarea networks typically are more detailed; requiring accurate coding of the number of lanes, turn pockets, merging lanes, turn restrictions and lane usage by time of day. Detailed information about traffic signals and transit schedules is also desirable.

Although the Alexandria case study includes all of the streets within the city of Alexandria, the study is being simulated as if it were a regional network. Synthetically generated pocket lanes and traffic controls are used by default.

2.2 Achieving User Equilibrium

TRANSIMS applications typically use a series of feedback loops between the Microsimulator and the Router to adjust travel paths based on simulated speeds. This iterative process stabilizes the travel demand with the network's ability to accommodate traffic by time of day. In theory this process continues until each traveler cannot improve their total travel time by changing paths. In practice user-equilibrium is typically defined when only two percent of the travelers can improve their travel time.

The user equilibrium process begins by re-routing all travelers using the latest simulated travel times. The **PlanCompare** program is then used to compare the total travel time for each traveler with the total travel time stored in the previous plan file. A percentage of those travelers with significantly different travel times are selected for inclusion in the full plan set input to the next simulation. The process continues until the number of travelers with meaningful travel time differences is small (about 2 percent).

3.0 How to Simulate Travel Plans on a TRANSIMS Network

This section describes the process for preparing input data, running the simulation, and generating output data with the Microsimulator.

3.1 Input Data Files

The Microsimulator requires a set of network files, a time-sorted plan file, and a vehicle file. The network files will have been synthetically generated as part of the Highway Network Conversion How-To and the Transit Network Conversion How-To. The plan file will have been generated by the last iteration of the Router feedback process described in section 5.5 of the Router and Router Feedback How-To. The vehicle file, which defines the type and location of each vehicle on the network, is one of the outputs of the Trip Table Conversion How-To.

The travel plans that are input to the Microsimulator need to be sorted by time of day using the **PlanPrep** program. A sample control file, named "6.Alex.2005.Trip.PlanSort.ctl," is provided in the control directory. This file is a text file that can be reviewed and edited using a standard text editor. The file contains the following records:

```
TITLE                      Sort the Plan Files for Simulation 6.Alex
DEFAULT_FILE_FORMAT        TAB_DELIMITED
PROJECT_DIRECTORY          ../

INPUT_PLAN_FILE            demand/6.Alex.2005.Trip.TravelPlans
OUTPUT_PLAN_FILE           demand/6.Alex.2005.Trip.TimePlans
PLAN_SORT_OPTION           TIME
```

The **PlanPrep** program can be executed using one of the batch files included in the control directory:

TripModel.bat (Windows)

This batch will run the full model case. However, a new batch file that only runs through the **PlanPrep** execution can be created by deleting the steps which follow the **PlanPrep** call that runs the control file 6.Alex.2005.Trip.PlanSort.ctl. A printout file, "6.Alex.2005.Trip.PlanSort.prn," and a new plan file, "6.Alex.2005.Trip.TimePlans" will be created by the process. The plan file is then fed into the Microsimulator.

3.2 Microsimulator Control File

A Microsimulator control file, "6.Alex.2005.Trip.Msim.ctl", is provided in the control directory. It is a text file that can be reviewed and edited using a standard text editor. Some of the more important control keys of the program are listed below.

```
TITLE                              Simulate the Trips for 6.Alex
DEFAULT_FILE_FORMAT                TAB_DELIMITED
PROJECT_DIRECTORY                  ../

NET_DIRECTORY                      ../network
NET_NODE_TABLE                     Node
NET_LINK_TABLE                     Link
NET_POCKET_LANE_TABLE              Pocket_Lane
NET_PARKING_TABLE                  Parking_2
NET_LANE_CONNECTIVITY_TABLE        Lane_Connectivity
NET_ACTIVITY_LOCATION_TABLE        Activity_Location_3
NET_PROCESS_LINK_TABLE             Process_Link_2
NET_UNSIGNALIZED_NODE_TABLE        Unsignalized_Node
NET_SIGNALIZED_NODE_TABLE          Signalized_Node
NET_TIMING_PLAN_TABLE              Timing_Plan
NET_PHASING_PLAN_TABLE             Phasing_Plan
NET_DETECTOR_TABLE                 Detector
NET_SIGNAL_COORDINATOR_TABLE       Signal_Coordinator
NET_TRANSIT_STOP_TABLE             Transit_Stop
NET_TRANSIT_ROUTE_TABLE            Transit_Route
NET_TRANSIT_SCHEDULE_TABLE         Transit_Schedule
NET_TRANSIT_DRIVER_TABLE           Transit_Driver

VEHICLE_FILE                       demand/Alex.2005.Trip.Vehicles
SORT_VEHICLES                      TRUE
VEHICLE_TYPE_FILE                  Inputs/Vehicle_Type.txt

PLAN_FILE                          demand/6.Alex.2005.Trip.TimePlans
NODE_LIST_PATHS                    Yes

CELL_SIZE                          7.5        //---- meters ----
TIME_STEPS_PER_SECOND              1          //---- steps / second ----
TIME_OF_DAY_FORMAT                 24_HOUR_CLOCK
SIMULATION_START_TIME              0:00
SIMULATION_END_TIME                26:00

SPEED_CALCULATION_METHOD           CELL-BASED

DRIVER_REACTION_TIME               0.7, 0.8, 0.9, 1.0
PERMISSION_PROBABILITY             55         //---- percent ----
SLOW_DOWN_PROBABILITY              10, 15, 20 //---- percent by facility type ----
SLOW_DOWN_PERCENTAGE               10, 20     //---- percent by facility type ----
RANDOM_NUMBER_SEED                 1623

PLAN_FOLLOWING_DISTANCE            525        //---- meters ----
LOOK_AHEAD_DISTANCE                260        //---- meters ----
LOOK_AHEAD_TIME_FACTOR             1.0        //---- imped / second ----
LOOK_AHEAD_LANE_FACTOR             4.0        //---- imped / lane change ----
MAXIMUM_SWAPPING_SPEED             37.5       //---- meters / second ----
```

```
MAXIMUM_SPEED_DIFFERENCE                 7.5         //---- meters / second ----
ENFORCE_PARKING_LANES                    Yes

MINIMUM_WAITING_TIME                     60          //---- seconds ----
MAXIMUM_WAITING_TIME                     120         //---- seconds ----
MAX_ARRIVAL_TIME_VARIANCE                 60         //---- minutes ----
MAX_DEPARTURE_TIME_VARIANCE               60         //---- minutes ----

NEW_PROBLEM_FILE                         results/6.Alex.2005.Trip.MsimProblems
MAX_SIMULATION_ERRORS                    100000


OUTPUT_SNAPSHOT_FILE_1                   results/6.Alex.2005.Trip.Snapshots
OUTPUT_SNAPSHOT_TIME_FORMAT_1            24_HOUR_CLOCK
OUTPUT_SNAPSHOT_INCREMENT_1               0:15
OUTPUT_SNAPSHOT_TIME_RANGE_1             7:00..9:00, 15:30..18:30

OUTPUT_OCCUPANCY_FILE_1                  results/6.Alex.2005.Trip.Occupancy_Max
OUTPUT_OCCUPANCY_TIME_FORMAT_1          24_HOUR_CLOCK
OUTPUT_OCCUPANCY_INCREMENT_1             1:00
OUTPUT_OCCUPANCY_TIME_RANGE_1            7:00..9:00, 15:30..18:30
OUTPUT_OCCUPANCY_MAX_FLAG_1              YES

OUTPUT_OCCUPANCY_FILE_2                  results/6.Alex.2005.Trip.Occupancy_Avg
OUTPUT_OCCUPANCY_TIME_FORMAT_2          24_HOUR_CLOCK
OUTPUT_OCCUPANCY_INCREMENT_2             1:00
OUTPUT_OCCUPANCY_TIME_RANGE_2            7:00..9:00, 15:30..18:30
OUTPUT_OCCUPANCY_MAX_FLAG_2              NO

OUTPUT_SUMMARY_FILE_1                     results/6.Alex.2005.Trip.Performance
OUTPUT_SUMMARY_TIME_FORMAT_1             24_HOUR_CLOCK
OUTPUT_SUMMARY_INCREMENT_1               0:15
OUTPUT_SUMMARY_TIME_RANGE_1              0:00..25:00
OUTPUT_SUMMARY_TURN_FLAG_1               YES

OUTPUT_SPEED_FILE_1                      esults/6.Alex.2005.Trip.Speed_Bins
OUTPUT_SPEED_FORMAT_1                    VERSION3
OUTPUT_SPEED_VEHICLE_TYPE_1              1:0
OUTPUT_SPEED_FILTER_1                    10
OUTPUT_SPEED_TIME_FORMAT_1               24_HOUR_CLOCK
OUTPUT_SPEED_INCREMENT_1                 0:15
OUTPUT_SPEED_TIME_RANGE_1                7:00..9:00, 15:30..18:30
OUTPUT_SPEED_LINK_RANGE_1                1..10
OUTPUT_SPEED_SAMPLE_TIME_1               2
OUTPUT_SPEED_BOX_LENGTH_1                40
OUTPUT_SPEED_NUM_BINS_1                  4

OUTPUT_EVENT_TYPE_1                      START_TIME, END_TIME
OUTPUT_EVENT_FILE_1                      results/6.Alex.Events
OUTPUT_EVENT_FILTER_1                    60
OUTPUT_EVENT_TIME_FORMAT_1               24_HOUR_CLOCK
OUTPUT_EVENT_TIME_RANGE_1                 0..86400

OUTPUT_RIDERSHIP_FILE_1                  results/6.Alex.2005.Trip.Riders
OUTPUT_RIDERSHIP_FORMAT_1                TAB_DELIMITED
OUTPUT_RIDERSHIP_TIME_FORMAT_1           24_HOUR_CLOCK
```

OUTPUT_RIDERSHIP_TIME_RANGE_1 7:00..9:00, 15:30..18:30

Control keys starting with "NET_" point to network files residing in the network directory. These files were created by the TransimsNet, IntControl, and TransitNet programs in the Highway Network and Transit Network Conversion How-Tos.

The time-sorted plan file, "6.Alex.2005.Trip.TimePlans," was generated by the **PlanPrep** program in Section 3.1. The "Vehicle" file, which contains the vehicle type and starting location of each vehicle in the network, was generated by the **ConvertTrips** program in the Trip Table Conversion How-To. The user-provided vehicle type file, "VehType.txt," contains information about the length, maximum speed, and acceleration and deceleration characteristics of each vehicle type.

The default value for `CELL_SIZE` is 7.5 meters, and the default value for `TIME_STEPS_PER_SECOND` is one second. The simulation starts at time 0:00 (midnight) and ends at 26:00 (2:00 AM the following day). A 26-hour period is used to enable vehicles that start trips close to midnight to complete their trips before the simulation concludes. The Microsimulator will automatically stop when the last vehicle trip has been completed.

The `MAXIMUM_WAITING_TIME` value of 120 seconds defines how long a vehicle must be stopped in one place before it is removed from the simulation (i.e., "lost" as discussed in Section 2.1). The `MAX_DEPARTURE_TIME_VARIANCE` and `MAX_ARRIVAL_TIME_VARIANCE` `of` 60 minutes means that any vehicle that is unable to be loaded to the network within 60 minutes after its scheduled departure time or that has not completed its trip within 60 minutes after its scheduled arrival time will be removed from the network. The `NEW_PROBLEM_FILE` "MsimProblem" will include the time, location, and traveler information for each of these events.

The vehicle lane-changing behavior is key to the validity of the simulation. This behavior is controlled by several parameters. The `PLAN_FOLLOWING_DISTANCE` controls how far before each turning movement a vehicle will start changing lanes in order to be in the required lane at the turn point. In this simulation, the key is set to 525 meters.

The three look-ahead parameters (`LOOK_AHEAD_DISTANCE, LOOK_AHEAD_TIME_FACTOR and LOOK_AHEAD_LANE_FACTOR)` control discretionary lane changing. The parameters define how far ahead a traveler will look in evaluating the option of changing lanes to move around slower traffic. In this simulation, the traveler will look ahead 260 meters and will value four seconds of travel time saved as comparable to one lane change maneuver.

`DRIVER_REACTION_TIME` defines the minimum car-following distance as a function of speed. In this simulation, vehicles will leave a gap equal to 0.7 seconds of travel time between themselves and the vehicle ahead of them.

3.3 Running the Microsimulator

The Microsimulator program can be executed using the batch files included in the batch directory:

TripModel.bat (Windows)

This batch will run the full model case. However, a new batch file that only runs through the first **Microsimulator** execution can be created by deleting the steps which follow the **Microsimulator** call that runs the control file 6.Alex.2005.Trip.Msim.ctl. The printout file "6.Alex.2005.Trip.Msim.prn" will be created by the process, as well as the output files discussed in the following section.

Summary statistics, such as the number of total plans read, the number of plans simulated, and the number and type of problems, can be found in the report file. You should review these values to ensure that the number of plans with problems is small enough that it does not significantly bias the results. If the number of plans with problems is significant, several Router feedback iterations will likely be needed to address the congestion-related problems. If serious problems persist, they are likely to be the result of a network coding error. Reviewing the network coding near the origin of a cascading queue is a necessary step in debugging and cleaning a TRANSIMS network. A description of the likely cause of several common problem messages can be found in Section 5.0.

3.4 Output Files

As their names imply, control keys preceded by the word "output" specify parameters controlling the output files generated by the Microsimulator. Three output files are defined for this application. They are typically referred to as "Snapshot," "Link Delay," and "Problem Link" files.

The Snapshot file is controlled by the `OUTPUT_SHAPSHOT_*` keys. It contains the position and speed of every vehicle in the simulation at specified time intervals. Snapshot files tend to occupy many megabytes, and can be as large as several gigabytes, depending on the number of vehicles simulated and the output time increment selected. For this reason, recording a second-by-second snapshot of the entire simulation may be impractical. Since the Alexandria network is relatively small, this application takes a snapshot every 15 seconds. As an option, you can specify more than one snapshot file, so as to focus on different areas with varying time increments. The Snapshot file is used by the Output Visualizer and the ArcSnapshot programs to display vehicle movements over time.

The Link Delay file contains time-dependent link flows and travel times. It is essentially the total volume and average travel time over a specified period of time (typically 15 minutes). This file is important because the information it contains about link travel times is used by the Router to re-route trips on the network. The `OUTPUT_SUMMARY_*` keys define the Link Delay file. This application saves 15-minute (900 second) summaries of link volumes and travel times. The `OUTPUT_SUMMARY_TURN_FLAG` key instructions the program to save turning movement volumes and delays as well.

The Problem file, defined by the `NEW_PROBLEM_FILE` key, contains information about the types of problems travelers experience on network links. The time, location, and type of problem encountered by each traveler are recorded in this file. Because the Problem file can be very large, it is often desirable to summarize the problems by network link in order to identify patterns and network errors. The problems can be summarized using the `OUTPUT_PROBLEM_*` keys. In this instance, the "6.Alex.2005.Trip.MsimProblems" file contains summary information about the locations where lane connectivity and wait time problems prohibit vehicles from completing their trip. The problems are aggregated into one hour totals (i.e., 3600 seconds) and only links with 100 or more problems are saved to the output file.

3.5 Viewing Links with Simulation Problems

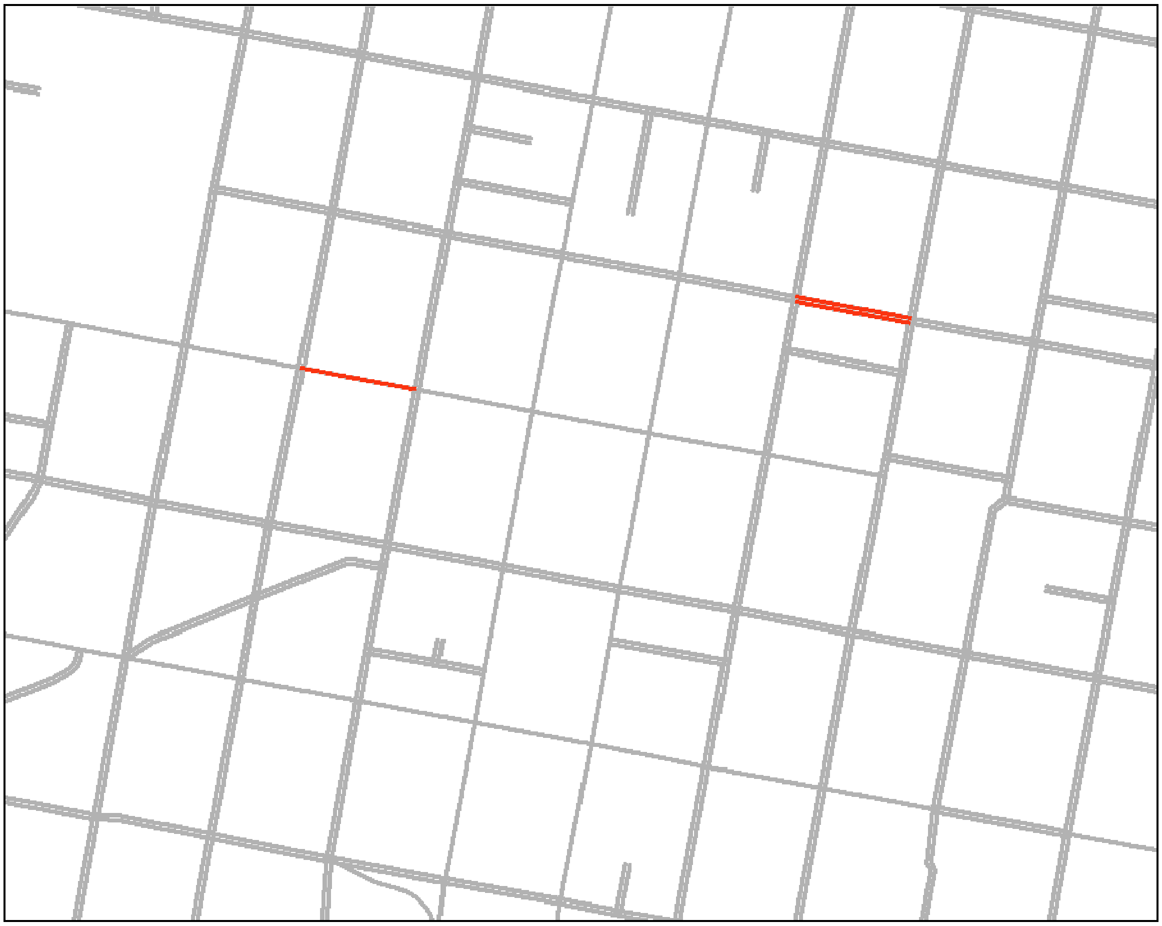
This section describes the option of using ArcGIS to highlight network links that have a significant number of simulation problems. It uses the Link shapefile created in the Highway Network Conversion How-To.

You should open ArcMap and add the "Link.shp" file, which can be found in the arcview directory of the network folder, as a new layer. You can then join the information from the "6.Alex.2005.Trip.MsimProblems" file to the network link using the following steps:

1. Right-click on the Link layer shown in the table of contents window in the left part of the screen and select "Joins and Relates" and then select "Join."
2. In the form shown select "Join attributes from a table" in the first list box.
3. Choose "LINK" in the second list box.
4. Click the folder button next to the third list box and point to the "6.Alex.2005.Trip.MsimProblems" file located in the results folder of the Alexandria directory. If ArcMap is not able to view the file, adding a "txt" extension to the file will resolve the problem.
5. Choose "LINK" from the bottom list box and press "OK."

This will add the fields of the "6.Alex.2005.Trip.MsimProblems" file to the attribute table of the Link shapefile. To view the added data, right-click on the Link layer and choose "Open Attribute Table."

In the figure below, red-colored links indicate locations where 100 or more vehicles were "lost" in the simulation. This can help identify locations where the network coding should be reviewed in more detail.



4.0 Summarizing the Microsimulator Output

The Microsimulator typically generates an immense amount of detailed information about the performance of the system. This information can be extremely useful in analyzing a specific area or a specific time of day. It can be difficult, however, to sift through the information to find what you need and to identify areas that require attention. A number of output summary tools, including those described here, can be used to manipulate the data and identify areas of interest.

4.1 LinkSum Program

The **LinkSum** program can be used to summarize link volumes, speeds, and volume-to-capacity ratios from the simulation output. Additional capabilities include reporting lane volumes and summing volumes for a user-defined group of links. This can help you obtain a macro-level view of the simulation results. The **LinkSum** program can also compare the "Link Delay" files from two Microsimulator runs to find differences in link volumes and travel times. This information can be useful in checking the convergence and stability of the equilibration process.

At a minimum, the required inputs to the **LinkSum** program are the Link Delay file generated by the Microsimulator and the Link file that contains the link characteristics of each link. You can specify which time periods to summarize and the summary time increment using the `SUMMARY_TIME_PERIODS` and `SUMMARY_TIME_INCREMENT` keys. In addition, links of separate facility types can be singled out for processing by defining the `SELECT_FACILITY_TYPES` key.

A sample LinkSum control file is provided in the control directory. The file “10.Alex.2005.Trip.LinkSum.ctl” is a text file that can be reviewed and edited using a standard text editor. The file records are listed below.

```
TITLE                                   Summarize the 10.Alex Simulation Results
PROJECT_DIRECTORY                        ../
DEFAULT_FILE_FORMAT                     TAB_DELIMITED

NET_DIRECTORY                             ../network
NET_NODE_TABLE                          Node
NET_LINK_TABLE                          Link
NET_LANE_CONNECTIVITY_TABLE             Lane_Connectivity
NET_ACTIVITY_LOCATION_TABLE              Activity_Location

LINK_DELAY_FILE                         results/10.Alex.2005.Trip.Performance
SUMMARY_TIME_PERIODS                    7:00..9:00, 15:30..18:30
SUMMARY_TIME_INCREMENT                  60          //---- minutes ----

NEW_LINK_DIRECTION_FILE_1               results/10.Alex.2005.Trip.Link_Volume
NEW_LINK_DIRECTION_FIELD_1              VOLUME

LINKSUM_REPORT_1                  TOP_100_VOLUME_CAPACITY_RATIOS
LINKSUM_REPORT_2                  TOP_100_LINK_VOLUMES
LINKSUM_REPORT_3                  NETWORK_PERFORMANCE_SUMMARY
```

The control key specifications defined above will generate a separate file containing 60-minute aggregate statistics of the volumes. See the LinkSum quick reference documentation for the other NEW_LINK_FILE_# files which can be saved such as speed, travel time, volume-to-capacity, etc. These files can be joined with the link file in ArcGIS so that the data can be displayed on maps.

In addition, summary reports are generated according to the `LINKSUM_REPORT_*` keys and included in the “LinkSum.prn” file. For example, based on the control configuration shown above, LinkSum will report the top-100 link volumes and the top-100 volume-to-capacity ratios.

4.2 Validate Program

The Validate program is used to compare the link flows generated by the Microsimulator with traffic counts. It can also be used to evaluate link flows generated by the Router stabilization process described in the Router and Router Feedback How-To. The program computes and reports the difference for every link for which there is a count, and it can generate performance statistics for groups of links. At a minimum, the required inputs to the Validate program are the Node and Link network files, a Link Delay file (the output from the Microsimulator), and a file containing counts on network links. Optionally, you can define screenlines by providing a file containing the list of links included in each screenline.

The Validate program reports several statistics, such as the difference between volumes and counts, average error, standard deviation, and root mean square error (RMSE). All of the statistical calculations made by the Validate program are limited to links with non-zero traffic counts and a specified time period. For more information on the complete set of statistical measures and their definition, refer to the Validate program documentation.

The sample control file shown below can be used to compare link flows from the Microsimulator to an arbitrary set of counts.

```
TITLE                          Alexandria Network Validation

#---- Input Files ----

NET_DIRECTORY                  ../network
NET_LINK_TABLE                 Link
NET_NODE_TABLE                 Node
LINK_DELAY_FILE                ../results/LinkDelay11
TRAFFIC_COUNT_FILE             ../network/Daily_Counts.txt
LINK_EQUIVALENCE_FILE    .     ./network/Link_Equiv.txt

#---- Parameters ----

SUMMARY_TIME_RANGE             0..24
SUMMARY_TIME_INCREMENT         0

VALIDATE_REPORT_1              LINK_GROUP_SUMMARY
VALIDATE_REPORT_2              FACILITY_TYPE_SUMMARY
```

In the configuration shown above, the control keys preceded by the NET keyword specify the location of the Node and Link files, the `LINK_DELAY_FILE` key defines the location of the Link Delay file generated by the Microsimulator. Daily traffic count information is contained in the "Daily_Counts.txt" file. The "Link_Equiv.txt" file defines the links included in link groups or screenlines. For more information about the format of these files, refer to the Validate program documentation. The `SUMMARY_TIME_RANGE` key is set to summarize link volumes contained in the Link Delay file for an entire day. The zero shown for the `SUMMARY_TIME_INCREMENT` key indicates that the time range will not be subdivided; in other words, one 24-hour time period is generated for each link direction.

The `VALIDATE_REPORT_*` keys identify the type of the summary reports that are generated. In this example, summary statistics for each screenline (i.e., link group) and facility type are requested.

Use case data for the Validate program is available in the 'Alexandria 2007-12-14' test data which can be downloaded from http://sourceforge.net/projects/transims/files/ → test data → Alexandria 2007-12-14 → Alexandria.zip. The control file Validate.ctl and the batch file Validate.bat can be used to run the **Validate** program assuming the full model has been executed using the RunEntireCase.bat script.

5.0 How to Improve Results through Microsimulator Feedback

This section describes how to feed the results of the Microsimulator back to the Router to adjust the travel plans for individual travelers. The feedback concept is similar to the one described in the Router and Router Feedback How-To. The primary difference is that instead of using the **PlanSum** program and BPR functions to calculate link travel times, this feedback process uses simulated travel times.

5.1 Rerouting Travelers

The process of selecting travelers for re-routing described in the Router and Router Feedback How-To can also be applied with the Link Delay file generated by the Microsimulator. This is often helpful in addressing congestion problems at specific locations or by time of day. Once these problems are resolved, a user-equilibrium convergence process will typically be initiated by re-routing all travelers and comparing their new trip travel times with the trip travel times calculated for the previous set of travel plans. The Router control file that re-routes the travelers based on the simulated travel times from the first Microsimulator run is named 6.Alex.2005.Trip.Router.ctl and is shown below.

```
TITLE                               Route the Highway and Transit Trips for 6.Alex
DEFAULT_FILE_FORMAT                 TAB_DELIMITED
PROJECT_DIRECTORY                   ../

NET_DIRECTORY                       ../network
NET_NODE_TABLE                      Node
NET_LINK_TABLE                      Link
NET_POCKET_LANE_TABLE               Pocket_Lane
NET_PARKING_TABLE                   Parking_2
NET_LANE_CONNECTIVITY_TABLE         Lane_Connectivity
NET_ACTIVITY_LOCATION_TABLE         Activity_Location_3
NET_PROCESS_LINK_TABLE              Process_Link_2
NET_TRANSIT_STOP_TABLE              Transit_Stop
NET_TRANSIT_ROUTE_TABLE             Transit_Route
NET_TRANSIT_SCHEDULE_TABLE          Transit_Schedule

HOUSEHOLD_LIST                      demand/5.Alex.2005.Trip.HH_List
LINK_DELAY_FILE                     results/5.Alex.2005.Trip.LinkDelay

TRIP_FILE                           demand/Alex.2005.Trip.Trips
TIME_OF_DAY_FORMAT                  24_HOUR_CLOCK
VEHICLE_FILE                        demand/Alex.2005.Trip.Vehicles

NEW_PLAN_FILE                       demand/6.Alex.2005.Trip.Plans
NEW_PROBLEM_FILE                    results/6.Alex.2005.Trip.Problems

NODE_LIST_PATHS                     YES
LIMIT_PARKING_ACCESS                YES
IGNORE_TIME_CONSTRAINTS             YES
```

```
WALK_PATH_DETAILS               YES
WALK_SPEED                      1.0       //---- meters / second ----
WALK_TIME_VALUE                 20.0      //---- imped / second ----
VEHICLE_TIME_VALUE              10.0      //---- imped / second ----
FIRST_WAIT_VALUE                20.0      //---- imped / second ----
TRANSFER_WAIT_VALUE             20.0      //---- imped / second ----
DISTANCE_VALUE                  1.0       //---- imped / meter ----
COST_VALUE                      5.0       //---- imped / cent ----
TRANSFER_PENALTY                1200      //---- impedance ----
MAX_WALK_DISTANCE               2000      //---- meters ----
MIN_WAIT_TIME                   60        //---- seconds ----
LEFT_TURN_PENALTY               300       //---- impedance ----
UTURN_PENALTY                   5000      //---- impedance ----
KISS_RIDE_STOP_TYPES            STOP, STATION, EXTERNAL
MAX_KISS_RIDE_DROPOFF_WALK      100       //---- meters ----
PARKING_HOURS_BY_PURPOSE        8.5, 2.5, 1.0, 1.0    //---- hours ----
```

You can execute this step by running the batch file found in the batch directory:

TripModel.bat (Windows)

This batch will run the full model case. However, a new batch file that only runs through the **Router** execution can be created by deleting the steps which follow the **Router** call that runs the control file 6.Alex.2005.Trip.Router.ctl. A printout file "6.Alex.2005.Trip.Router.prn" will be generated by the process, along with new plan and problem files. The feedback loop continues by identifying the plans to retain from this step.

5.2 Selecting Travel Plans for the Next Iteration

You could choose to simulate the new travel plans generated by the all-or-nothing routing using simulated speeds. It is often more efficient, however, to use only a fraction of these plans in the next simulation to avoid oscillation effects that hinder convergence. The selection process thus focuses on those travelers whose total trip travel time is significantly different from their previous travel plan. When the number of travelers with meaningful travel time differences becomes small (about 2 percent), the Router–Microsimulator process approximates a user-equilibrium solution.

The **PlanSelect** program is used to implement the travel time comparison and selection process. The control file used for this procedure is named 7.Alex.2005.Trip.PlanSelect.ctl and is shown below.

```
TITLE                           Select Travelers to Re-Route in 7.Alex
DEFAULT_FILE_FORMAT             TAB_DELIMITED
PROJECT_DIRECTORY               ../

VEHICLE_FILE                    demand/Alex.2005.Trip.Vehicles

PLAN_FILE                       demand/6.Alex.2005.Trip.TravelPlans
LINK_DELAY_FILE                 results/6.Alex.2005.Trip.Performance

NEW_HOUSEHOLD_LIST              demand/6.Alex.2005.Trip.HH_List
```

```
NET_DIRECTORY                         ../network
NET_NODE_TABLE                        Node
NET_LINK_TABLE                        Link
NET_PARKING_TABLE                     Parking_2
NET_LANE_CONNECTIVITY_TABLE           Lane_Connectivity


PERCENT_TIME_DIFFERENCE               10.0     //---- percent ----

MINIMUM_TIME_DIFFERENCE                1       //---- minutes ----
MAXIMUM_TIME_DIFFERENCE               60       //---- minutes ----
SELECTION_PERCENTAGE                  50       //---- percent ----
MAXIMUM_PERCENT_SELECTED              10       //---- percent ----
RANDOM_NUMBER_SEED                    755
```

Based on the control file shown above, the **PlanSelect** program will identify households whose travel time changed more than 10 percent from iteration 6 to iteration 7. This will be done by comparing travel times from in the current Plan file to link delays from the previous microsimulation. In addition, the `MINIMUM_TIME_DIFFERENCE` and the `MAXIMUM_TIME_DIFFERENCE` control keys specify that the absolute difference must be a least 1 minutes but no more than 60 minutes. The `SELECTION_PERCENTAGE` key designates that up to 50 percent of the selected plans will be saved to the output household list file, but no more than 10 percent of the total travelers will be selected. The values of the five keys described above have been found to lead to good starting point for the convergence process. After several iterations, the modeler may wish to tighten the selection criteria to include less significant travel time differences.

The utility **PlanCompare** can also be used to implement the travel time comparison and selection process. In this case plans are actually selected and retained instead of selecting households to re-route. In model applications conducted to date this approach has been found to yield a better Router-Microsimulator user equilibrium condition.

PlanCompare is not applied in the Alexandria_4.0.3a test data. However, this utility is applied with the Alexandria 2007-12-14 test data. The batch script RunEntireCase.bat runs 10 router feedback iterations and 5 microsimulator feedback iterations. **PlanCompare** is used in the microsimulator feedback. A sample control file used for this procedure PlanCompare11.ctl is shown below.

```
TITLE                         Compare Plan Travel Times
INPUT_PLAN_FILE                ../plans/TravelPlans11
COMPARE_PLAN_FILE             ../plans/Plans11
OUTPUT_PLAN_FILE              ../plans/RetainPlans11

PERCENT_TIME_DIFFERENCE       4
MINIMUM_TIME_DIFFERENCE       1
MAXIMUM_TIME_DIFFERENCE       30
SELECTION_PERCENTAGE          50
MAXIMUM_PERCENT_SELECTED      10

RANDOM_NUMBER_SEED            12345
```

Based on the control file shown above, the **PlanCompare** program will identify plans whose travel time changed more than 10 percent from iteration 10 to iteration 11. This will be done by comparing the Plan files of the two iterations. In addition, the MINIMUM_TIME_DIFFERENCE and the MAXIMUM_TIME_DIFFERENCE control keys specify that the absolute difference must be a least 2 minutes but no more than 30 minutes. The SELECTION_PERCENTAGE key designates that up to 50 percent of the selected plans will be saved to the output plan file, but no more than 10 percent of the total travelers will be selected. The values of the five keys described above have been found to lead to good starting point for the convergence process. After several iterations, the modeler may wish to tighten the selection criteria to include less significant travel time differences.

5.3 Merging Plan Files

The plans retained by **PlanCompare** or the new plans developed by re-routing a selected household list produced by using **PlanSelect** need to be merged with the full regional plan set to create the travel plans for the next simulation. The **PlanPrep** control file that merges plans is named PlanPrep11a.ctl and is shown below. It can be reviewed and edited by a standard text editor.

```
TITLE                   Merge Plan Files
INPUT_PLAN_FILE         ../plans/RetainPlans11
MERGE_PLAN_FILE         ../plans/TravelPlans11
OUTPUT_PLAN_FILE        ../plans/TravelPlans12
PLANPREP_REPORT_1       PERCENT_PATH_CHANGE
PLANPREP_REPORT_2       PERCENT_TRAVEL_TIME_CHANGE
PLAN_SORT_OPTION        TRAVELER
```

The PlanPrep control file that would merge plans assuming use of **PlanSelect** as opposed to **PlanCompare** can be viewed with the Alexandria_4.0.3a test case data in the file 7.Alex.2005.Trip.PlanMerge.ctl.

5.4 Sorting the Merged Plan File

During the merge process, **PlanPrep** sorts the plans by traveler ID. The merged plan file needs to be sorted by time of day before being input to the Microsimulator. Once the merge process is complete, the **PlanPrep** program should be used to sort the plans by start time. The **PlanPrep** control file that sorts the plans by start time is named PlanPrep11b.ctl and shown below. It can be reviewed and edited by a standard text editor.

```
TITLE                   Sort Plan File
INPUT_PLAN_FILE         ../plans/TravelPlans12
MERGE_PLAN_FILE         NULL
OUTPUT_PLAN_FILE        ../plans/TimePlans12
PLAN_SORT_OPTION        TIME
```

This process sorts the plans in the “TravelPlans12” file by start time and writes the results to the “TimePlans12” file. The output plan file, TimePlans12, is ready to be read by the Microsimulator in the 12th feedback loop.

5.5 Batch Feedback Loops

To demonstrate how the Router-Microsimulator feedback process can be automated into a batch processing script, an iteration convergence process is provided. This process executes the Microsimulator, Router, PlanCompare, and PlanPrep programs for the 11th to the 15th iteration with the Alexandria 2007-12-14 test case data. The feedback loops can be run by using the following batch file:

RunEntireCase.bat (Windows)

You should review the printout files generated by the PlanCompare program to monitor the convergence process. The “Number of Travelers Selected” can be interpreted as the convergence error for each run. The process implemented by the batch file brings this convergence error down from 21 percent to 10 percent. A TRANSIMS application is typically considered in equilibrium when the percentage selected is two percent or less.

To demonstrate how the Router-Microsimulator feedback process can be automated into a batch processing script, a somewhat different iteration convergence process is also provided. This process executes the Microsimulator, Router, PlanSelect, and PlanPrep programs for the 7th to 10th iteration in the Alexandria_4.0.3a test case data. The feedback loops can be run by using the following batch file:

TripModel.bat (Windows)

6.0 Troubleshooting

The Microsimulator reads the plan file, but processes only a few trips.

Check if you have sorted the plans by start time before running the Microsimulator. If you have not, the Microsimulator will only simulate the first few vehicles.

What is a “Vehicle Access” problem?

This error message is generated when the Microsimulator is not able to find the assigned vehicle for the trip in the parking lot attached to the starting activity location or when the vehicle exists but its vehicle type is undefined. The user should check that the Vehicle Type file defines vehicle types for all the vehicles in the Trip file.

What is a “Wait Time Problem”?

A wait time problem is generated when a vehicle remains in the same cell unable to advance for an amount of time greater than the `MAX_WAIT_TIME` key. The most frequent cause of this problem is excessive congestion. It can also be caused by incorrect signal coding that does not provide a phase for all the eligible movements.

What is a “Parking Access Problem”?

This problem message is generated when a vehicle is not able to move from the parking lot to the first link in its journey. You should check that the link does not restrict vehicles of the particular vehicle type from using the link.

What is a “Departure Time Problem”?

When a vehicle cannot start its trip at the time specified in the trip file plus the amount of slack time defined by the `MAX_DEPARTURE_TIME_VARIANCE`, a departure time problem is generated. The most frequent cause of this problem is excessive congestion close to the starting parking lot, which prevents the vehicle from being loaded onto the first link.

What is an “Arrival Time Problem”?

If a vehicle is still traveling at the time it is scheduled to arrive at its destination plus the slack time defined by the `MAX_ARRIVAL_TIME_VARIANCE` key, it will be removed from the network, and an arrival time problem error will be generated. The most frequent cause of this type of problem is congestion. You can allocate more time for the vehicle to finish its trip by adjusting the value for the `END_TIME_CONSTRAINT` key in the Router.

What is a “Lane Connectivity Problem”?

This message is generated when no lane connectivity exists between two successive links of the vehicle’s path. This most often occurs when the network has been changed, but the travel plans have not been rebuilt. You should inspect the location generating the problem and restore the lane connectivity or re-route the traveler.

7.0 Frequently Asked Questions

How can the Microsimulator processing time be reduced?

Microsimulator run times depend heavily on the duration of the simulation, traffic congestion, the time step chosen, and the number of vehicles simulated. If the simulation is highly congested, it is often better to stop the Microsimulator after congestion sets in and re-route trips during the congested time period rather than let the Microsimulator churn away under gridlock conditions.

If congestion is not the problem, the slowdown may be due to excessive output processing. Writing detailed snapshot files, which can be many gigabytes, slows down the program significantly. Limiting the output to specific times of day or to specific geographical areas can noticeably improve performance. Using binary files can reduce the time for the input and output operations by half.

How many iterations do we need to achieve convergence?

The number depends on the congestion level, the convergence criterion used, and the method selected for achieving it. Studies in Portland and Washington, D.C., have shown that after approximately 30 Router–Microsimulator feedback loops, the percentage of travelers who could improve their travel time by changing paths is less than two percent. Additional iterations do not improve this result significantly.

How can Microsimulator fidelity be increased?

You should focus on the several parameters that control vehicle spacing, random speed fluctuations, and lane changing. Reducing the cell size has the potential to increase the fidelity of the simulation, provided the parameters are properly calibrated. Simple tests can be performed to estimate the impact of Microsimulator parameters on flow-density-speed relationships. It is also advisable to visualize vehicle movements through critical intersections and freeway weaving sections to verify the simulation results. Additional checks can include plotting of vehicle speeds and validating travel times against field data.