Dynamic User Equilibrium – TRANSIMS Version 5

Version 1.0 Revision History: September 2013 – Created by the Volpe Center

Document Background and Purpose

This is an update of the "Dynamic User Equilibrium How-To," prepared by RSG in April 2010. This document explains how TRANSIMS Router (version 5) can be used to achieve a dynamic user equilibrium (D.U.E.) convergence.

Table of Contents

1. Introduction

TRANSIMS is a suite of tools (listed in section 6 on page 13) that can be combined in a variety of ways to model a wide range of transportation scenarios. The overall software package emphasizes time-dependent path building and multi-modal network simulation. It also includes numerous tools for data conversion, processing, and analysis of model results.

TRANSIMS implements a model of person trips made by households and vehicle trips made by other entities between link-based activity locations, using specified travel modes, on a second by-second basis during the course of a 24-hour time period. The procedures described in this document use iterative feedback between model components. Within each feedback iteration, various tools may be used to adjust the activity patterns, activity locations, travel modes, travel schedules, and/or travel paths of a select number of individual travelers. Stopping criteria are based on network performance and traveler-based equilibrium convergence concepts.

The typical modeling process employs dynamic traffic assignment concepts using micro-simulated vehicle movements and time-dependent network attributes (e.g., traffic signals, lane-use and parking restrictions, parking and toll costs). The Router generates minimum impedance paths through the network for individual travelers (travel plans) and calculates performance measures based on link volume/delay functions. The Simulator implements the plans and calculates the system performance based on simulated vehicle queuing.

The Simulator is a capacity constrained process by nature, so it is helpful if the travel plans generated by the Router and input into the Simulator are less than or equal to the capacity of each roadway. Over loading the network causes cascading queues that can make the simulation results totally unrealistic. It also takes considerably more iterations for the Router-Simulator process to recover from gridlock conditions that typically occur when travel plans from an initial Router run are used directly with the Simulator.

An effective way of initializing the network performance and creating a set of travel plans that the Simulator can reasonably simulate is to apply a Router Stabilization process before attempting to stabilize simulated network performance and subsequently iterating to a dynamic user equilibrium solution. In other words, a typical TRANSIMS assignment process involves three iterative phases: Router Stabilization, Simulator Stabilization, and Dynamic User Equilibrium Convergence[1].

The Router Stabilization phase develops and distributes the travel paths on roadways based on traditional volume-delay functions like the Bureau of Public Roads (BPR) formula. Other volume-delay functions such as conical or exponential functions can also be selected by the user. Once the trips are distributed and the volume-to-capacity ratios are within reasonable ranges (i.e. less than 1.2), the second iterative feedback phase begins.

[1] See “Dynamic Traffic Assignment, A Primer,” Transportation Research Circular, no. E-C153, June 2011, and available at http://onlinepubs.trb.org/onlinepubs/circulars/ec153.pdf

This Simulation Stabilization phase focuses on reconciling the path travel times produced by the Simulator with those produced by the Router. This phase involves targeted re-routing of trips traveling through congested areas of the network, manual network refinements to improve the fidelity of the modeled network (e.g., pocket lanes, lane connectivity, transit schedules, etc.) and adjustments to signal timing plans based on intersection demand. Feedback iterations continue until nearly all travelers are able to complete their trips in a reasonable amount of time.

The Dynamic User Equilibrium (D.U.E.) condition is achieved when all trips for a particular origin-destination pair and time increment are on their shortest impedance routes. The Dynamic User Equilibrium phase therefore focuses on minimizing the travel times or generalized costs of individual travelers by comparing the path used in the routing or simulation to the minimum impedance path based on the network performance from the simulation and rerouting and simulating plans that can be significantly improved.

Figure 1 illustrates the typical three-phase TRANSIMS model simulation sequence.

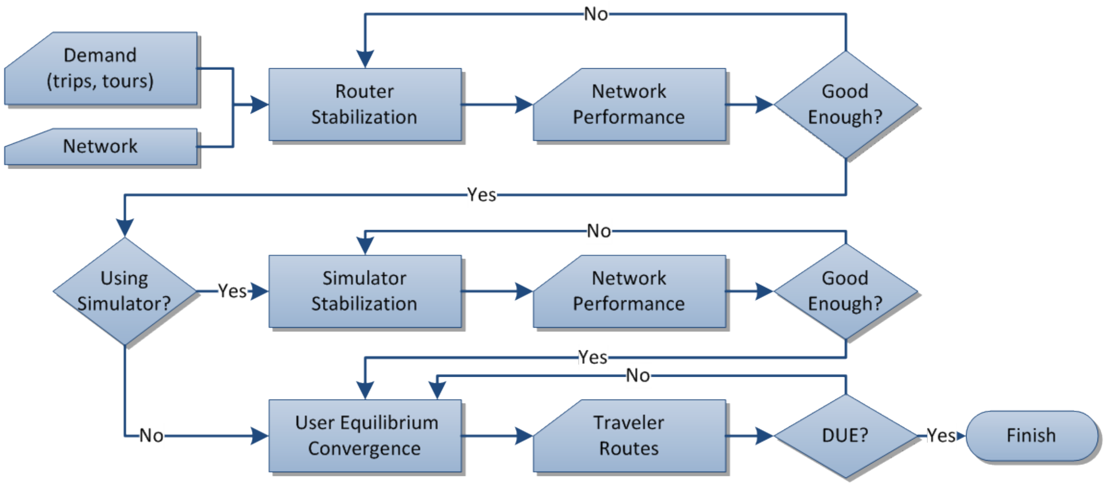

Figure 1 Three phase TRANSIMS simulation sequence

It should be noted that the use of the Simulator is entirely optional. It is possible to develop a Router-only user equilibrium model using volume-delay functions to estimate link travel times rather than the Simulator.

The following sections describe each phase in more detail and illustrate the primary TRANSIMS tools used in each. A brief description of the task performed by each tool is also provided.

This document assumes you have installed TRANSIMS Version 5.0 on a Windows or Linux computer system and that you understand the basic procedures and terminology for executing TRANSIMS programs.

The TRANSIMS software can be downloaded from http://sourceforge.net/projects/transims/. The Sourceforge site includes source code for version 4 and version 5, as well as executables for version 4. Documentation is available at http://code.google.com/p/transims/ .

Text files are used to store the input and output information. You need to be able to review and edit these files using a standard text editor (e.g., vi, NotePad, NotePad++, WordPad) or other software that can manipulate tab-delimited files (e.g., Excel).

2. Representing Supply and Demand in TRANSIMS

Before running a Router Stabilization routine it is assumed that TRANSIMS highway and/or transit networks have been developed using the TransimsNet, TransitNet and IntControl utilities. Please refer to the Highway Network How-To and Transit Network How-To and corresponding user manuals for more detailed information on these processes. Table 1 lists the network files, including information on whether they are required (Req) or optional (Opt) for the router and simulator.

Table 1 Network Files

Network Files	Router	Simulator	Comment
NODE_FILE	Req	Req	Required by both Router and Simulator
LINK_FILE	Req	Req	Required by both programs. Router uses cost and capacity information.
CONNECTION_FILE	Req	Req	Required by both programs. Simulator is more sensitive to the correctness of this file, particularly for multiple turn lanes.
LOCATION_FILE	Req	Req	Required by both programs.
POCKET_FILE	Opt	Req	Only required by Simulator.
PARKING_FILE	Opt	Req	Only required by Simulator.
LANE_USE_FILE	Opt	Opt	Not required, but is essential if lane use restrictions exist.
TURN_PENALTY_FILE	Opt	Opt	Not required, but is essential if turn restrictions or penalties exist.
ACCESS_FILE	Opt	Opt	Not required.
SIGN_FILE	Unused	Opt	Not used by the Router, but the Simulator depends on the correct representation of signs and signals.
SIGNAL_FILE	Unused	Opt	
PHASING_PLAN_FILE	Unused	Opt	
TIMING_PLAN_FILE	Unused	Opt	
DETECTOR_FILE	Unused	Opt	
TRANSIT_STOP_FILE	Opt	Opt	Used if transit is part of the model.
TRANSIT_FARE_FILE	Opt	Opt	
TRANSIT_ROUTE_FILE	Opt	Opt	
TRANSIT_SCHEDULE_FILE	Opt	Opt	
TRANSIT_DRIVER_FILE	Opt	Opt	

It is also assumed that data files of regional travel demand have been developed using TRANSIMS trip and tour-based tools. Developing activity-based demand requires the use of an existing activity-based demand model. Other TRANSIMS applications use trip-based zonal travel demand matrices extracted from a regional travel demand model. The process for converting zonal trip matrices into trip lists that can be used in TRANSIMS is described in the Trip Table Conversion How-To and ConvertTrips user manual.

3. Router Stabilization

As described in the introduction, the Router Stabilization process is used to effectively provide a "warm-start" for the subsequent application of the Simulator. The underlying concept is that volume-delay functions can first be used to establish and stabilize travel plans and network travel times before using the Simulator, which has a relatively long run time compared to the Router. In essence the Router Stabilization is used to distribute traffic in a logical fashion using traditional planning methods such as the BPR volume / delay functions prior to initiating the microsimulation.

Figure 2 illustrates the initial Router run, which produces an initial Link Delay and Travel Plans. Figure 3 illustrates the typical Router Stabilization routine which utilizes the TRANSIMS Router and PlanSelect utilities. This is run several times, with the iteration number X, being incremented by 1 each time.

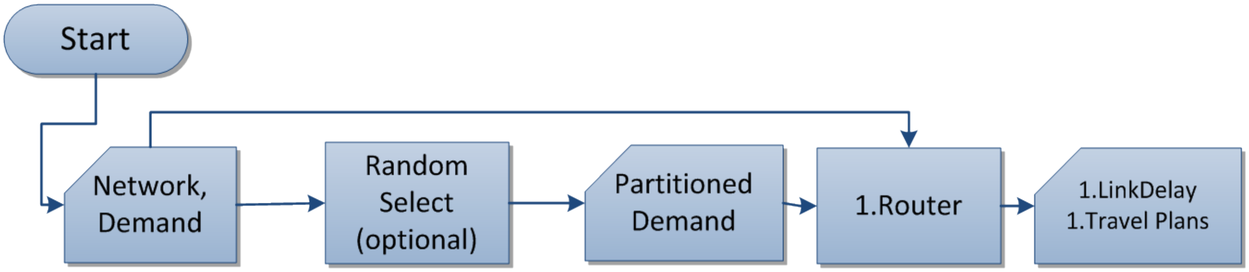

Figure 2 Router Initial Run

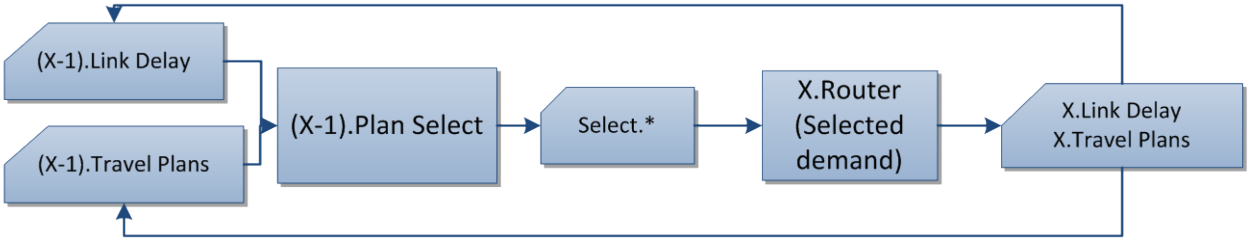

Figure 3 Router Stabilization

In the first iteration the Trip File is input to the Router to develop the set of outputs called 1.TravelPlans where the all-or-nothing assignment uses free-flow travel times. The plan file (a list of links traversed in the path) contains the origin, destination, trip start time, trip end time, and trip path for each household traveler. In the Router Stabilization phase, the PlanSelect utility is used to identify which travelers should be re-routed based on user-defined selection criteria. In this case, the re-routing criteria could be to select household travelers who traverse links with a V/C ratio, say, greater than 1.2.

In the second iteration, the Router develops plans using link travel times from the first iteration. However, only a sub-selection of household travelers are routed, so the resulting New Plans set is smaller than the full set of regional plans in the file Travel Plans. PlanSelect is once again executed to identify those travelers still traversing links with V/C ratios greater than 1.2. As the iterative process continues, fewer and fewer households should be selected and the Travel Plans file will stabilize and require fewer and fewer updates as the number of travelers satisfying the selection criteria is reduced.

TRANSIMS implementations conducted to date have demonstrated that using a selection criteria set that is modified during the iterative process most efficiently reduces the number of selected travelers. In the example illustrated below (Table 2), iterations 2 through 4 use a V/C selection criteria of 2.0, iterations 5 through 10 use a V/C selection criteria of 1.5, and finally iterations 11 through 15 use a travel time difference where current iteration travel times are compared to travel times from the previous iteration. In all cases, the **maximum_percent_selected** variable indicates that 50% of those trips meeting the criteria are selected. The **selection_percentage** variable indicates that the number of trips selected is limited to less than 10% of all trips. These limits are intended to prevent oscillation in the router stabilization process, where large numbers of trips are shifted between two roads, each road alternately becoming congested.

Table 2 Router Stabilization

Variables[2]	Iterations		
	2 - 4	5 - 10	11 - 15
Select_VC_Ratios	2.0	1.5	-
Percent Time Difference	-	-	10
Minimum Time Difference	-	-	2
Maximum Time Difference			45
Selection_percentage	50	50	50
Maximum_percent_selected	10	10	10
Select_time_periods	all	all	all

[2] See the TRANSIMS Parameter Reference for the definitions of these variables.

4. Microsimulation Stabilization

The Router Stabilization phase produces a set of Travel Plans that can be used as input in the Simulator. However, the travel times derived by use of the volume-delay functions in the Router Stabilization phase may be different than the travel times derived using the Simulator to simulate second-by-second vehicle movements throughout the day. Unlike the Router, the Simulator considers intersection delays and queuing. Therefore, another iterative process is required to reconcile the travel time estimates produced by the Router with the higher-fidelity travel time estimates produced by the Simulator. This process re-routes travelers whose trip duration time in the Travel Plan file is significantly different from the travel time calculated from the Simulation. It also identifies those travelers who fail to complete their trips in the Simulator (often due to excessive delay caused by intersection-related bottlenecks not considered in the Router). Figure 4 illustrates a simplified version of the Simulator stabilization process. Selected travelers are routed to produce a set of plans. Those plans are then sorted by time, and subsequently fed to the Simulator. The Simulator reports new link delays (the Performance file), as well as travelers that could not complete their simulated trips. These travelers are listed in a "Problem" file, and are then fed back to the router for re-routing. Additionally, the simulator link delays are averaged with the router link delays to produce new link delays that the router will consider when re-routing.

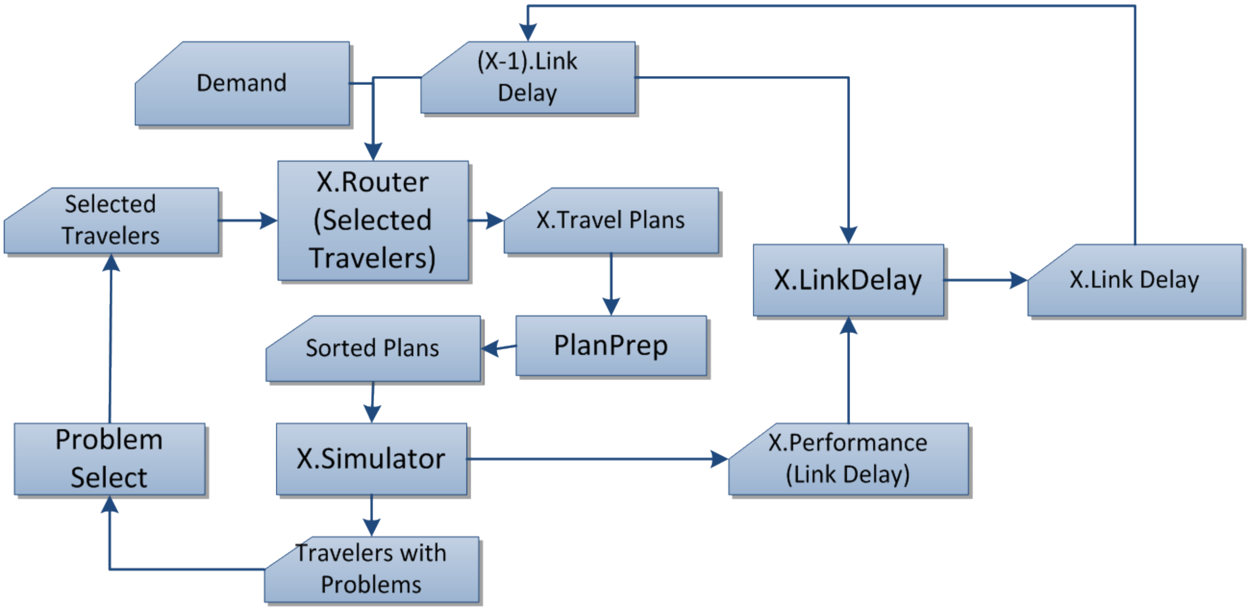

Figure 4 Simulator Stabilization (simplified)

In general, the iterative Microsimulation Stabilization scheme is comparable to the Router Stabilization scheme. However, in this case the Simulator is used to produce link volumes and times/costs. Another substantive difference in the Microsimulator Stabilization is the use of the LinkDelay utility. The LinkDelay program is used to average the link delays from each iteration to implement a Method of Successive Averages (MSA) approach, which is a simulation control strategy used to help iterative models move more quickly towards a converged solution

In the Microsimulation Stabilization, some different utilities are executed to select travelers for re-routing. Note that in the Router Stabilization process, the utility PlanSelect is used to identify which household travelers should be re-routed based on a user-specified selection criteria such as V/C ratios and/or travel time differences. PlanSelect may also be used to perform the traveler selection in the Microsimulator Stabilization feedback. However, the current best practice techniques use the utilities EventSum and ProblemSelect which read outputs from the Microsimulator to identify the household travelers for re-routing.

Figure 5 illustrates a more complicated Simulator Stabilization routine which utilizes the TRANSIMS Router, PlanPrep, PlanTrips, Simulator, LinkDelay, EventSum, and ProblemSelect utilities. (This figure was drawn from the TRANSIMS version 4 documentation, and therefore uses the older term "Microsimulator.")

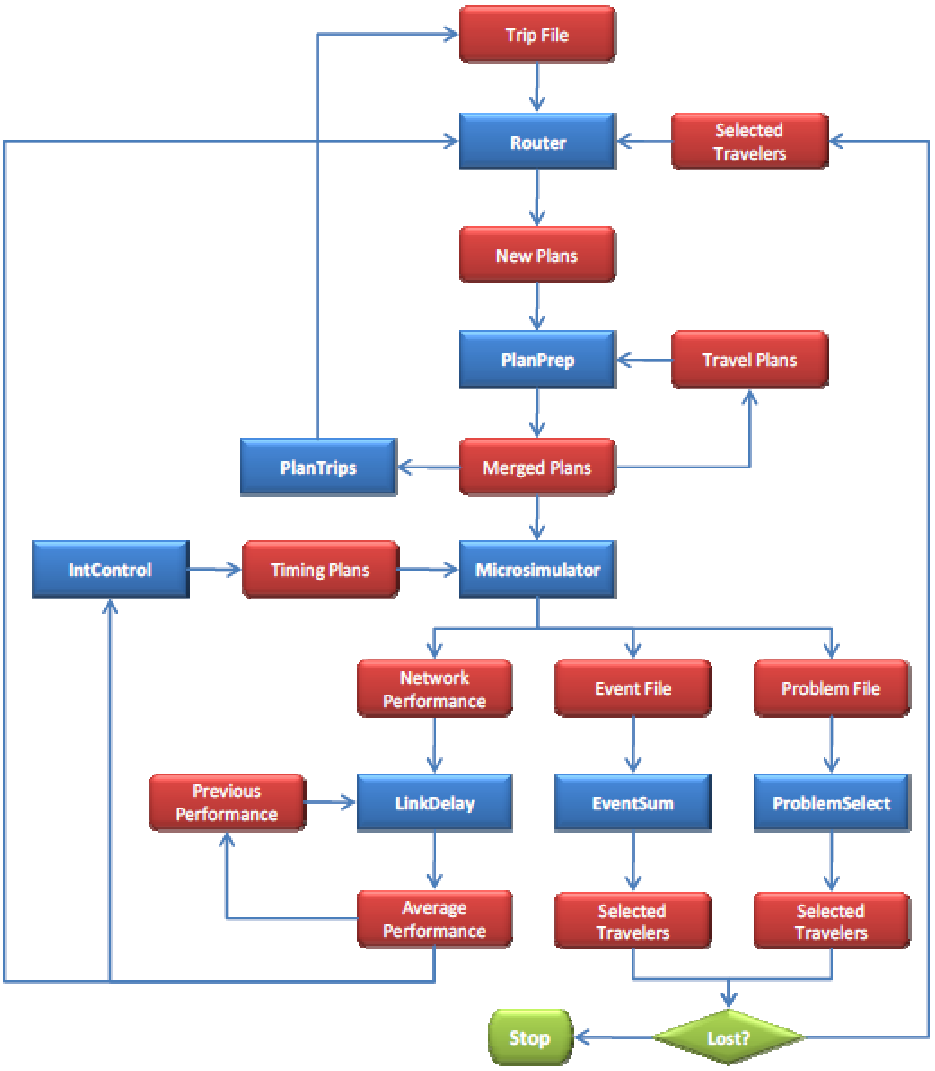

Figure 5 Microsimulation Stabilization

As with the Router Stabilization process, the Microsimulator Stabilization process iterates until an acceptably small number of households are selected by either EventSum, ProblemSelect and/or PlanSelect. In most implementations conducted to date, attempts are made to iterate the Microsimulator Stabilization until very few households and/or problems are selected, or such as less than 1% of the total household travelers.

It should be noted that the Simulator is much more sensitive to network fidelity than the Router. Likely problem areas include the location and timing of signals, lane connectivity (in particular, multiple turn lanes), and pocket lanes.

4.1 Link-Based Relative Gap

This is a link-based measure that compares the vehicle hours of travel for actual and minimum impedance paths. It may be helpful for evaluating Router or Simulator stabilization, and is typically expressed as

$$Relative\ Gap = \frac{|\sum VE_t \times CE_t - \sum VA_t \times CE_t|}{\sum VA_t \times CE_t}$$

where:

$\Sigma = summation\ over\ all\ network\ links$

$VE_t = the\ current\ volume\ on\ a\ given\ link\ and\ time\ increment$

$CE_t = the\ travel\ cost\ (time)\ associated\ with\ volume\ VE_t$

$VA_t = the\ link\ volume\ from\ an\ AON\ assignment\ based\ on\ CE_t$

In the TRANSIMS LinkSum program, the calculation is slightly different.

$$Relative\ Gap = \frac{\sum VE_t \times CE_t - \sum VA_t \times CA_t}{\sum VA_t \times CA_t}$$

$\Sigma = summation\ over\ all\ network\ links$

$VE_t = the\ previous\ volume\ on\ a\ given\ link\ and\ time\ increment$

$CE_t = the\ travel\ cost\ (time)\ associated\ with\ volume\ VE_t$

$VA_t = the\ link\ volume\ from\ an\ AON\ assignment\ based\ on\ CE_t$
$CA_t = the\ travel\ cost\ (time)\ from\ an\ AON\ assignment\ based\ on\ CE_t$

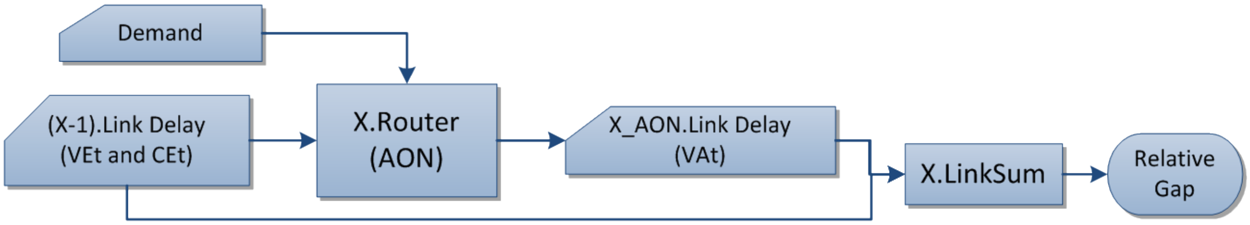

Figure 6 Calculation of Link-Based Relative Gap

5. Dynamic User Equilibrium

The third and final phase, Dynamic User Equilibrium (D.U.E.), focuses on minimizing the travel times or generalized costs of individual travelers by comparing the path used in the router or simulation to the minimum impedance path based on the network performance from the router or simulation and using this information to determine convergence.

5.1. Trip-Based Relative Gap

A trip-based relative gap measure is used to assess convergence[3]. This disaggregate measure is calculated at the trip path level, and using the same set of link impedances captures the difference between the trip cost using the most recent trip path and the trip cost using the shortest path. This difference is summed across all trips and normalized. Note that for each trip, the trip cost depends on the trip departure time, because trip cost is derived from time-dependent link costs.

$$Trip\ Gap = \frac{|\sum(CR_x\{C_{mt}\} - CA_x\{C_{mt}\})|}{\sum CA_x\,\{C_{mt}\}}$$

$where$:

$\Sigma = summation\ over\ all\ trips$

$\{C_{mt}\} = current\ time\ varying\ link\ costs$

$CA_x = AON\ cost\ of\ trip\ x\ based\ on\ link\ costs\ \{C_{mt}\}$

$CR_x = reskimmed\ costs\ for\ trip\ x\ along\ the\ path\ used\ to\ generate\ \{C_{mt}\}$

5.2. Implementation in TRANSIMS

Figure 7 illustrates an implementation of the User Equilibrium process in TRANSIMS, using the Router only (Macroscopic D.U.E.).

[3] See page 11 of the DTA Primer

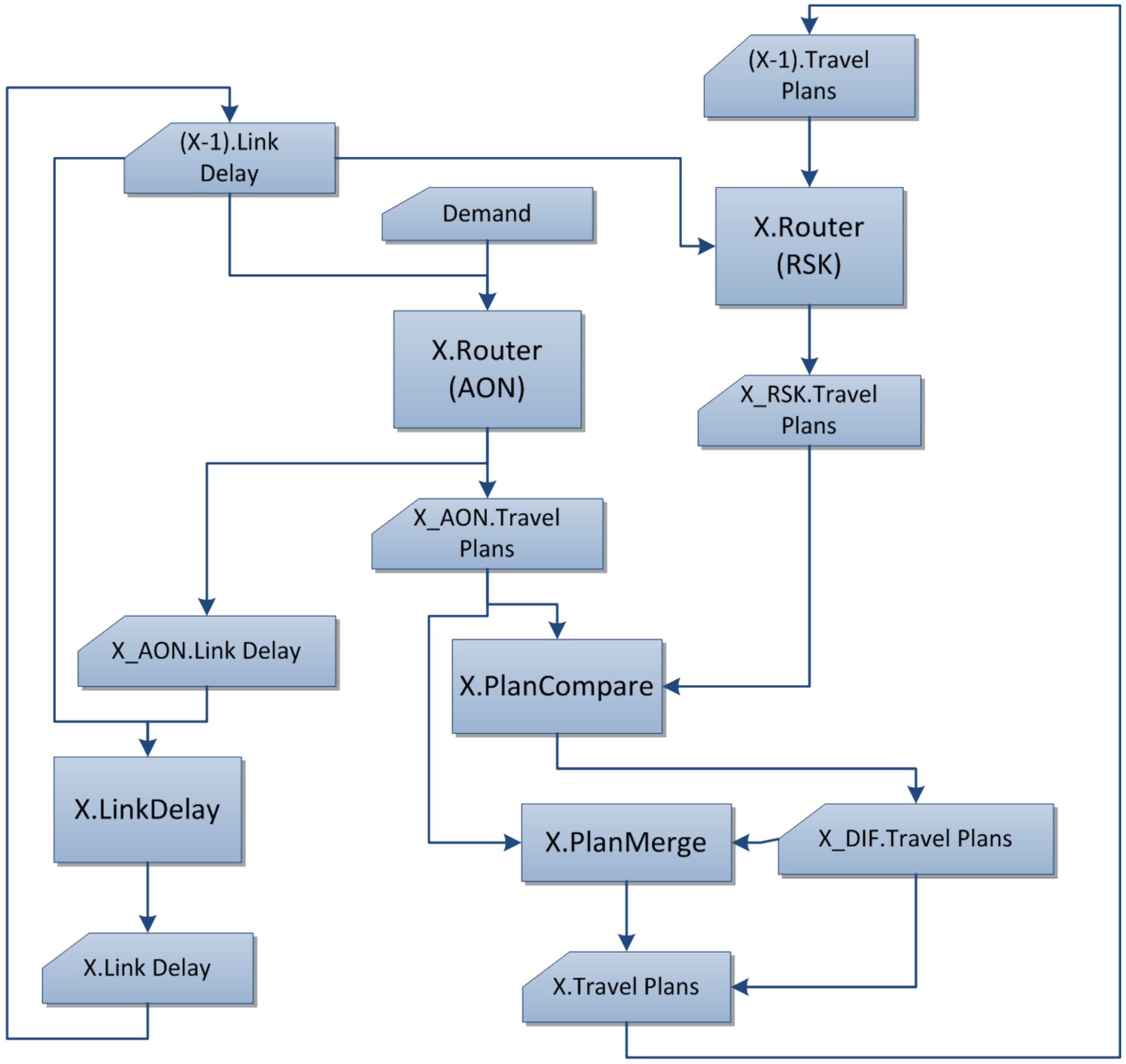

Figure 7 User Equilibrium Process (Router Only)

First, an All or Nothing (AON) assignment is computed by the router [X.Router (AON)], using all of the trips in the trip file [Demand] and the link travel times from the previous iteration [(X-1).Link Delay]. This produces a new set of AON travel plans [X_AON.Travel Plans], and a new set of link flows and travel times {X_AON.Link Delay].

Second, the travel plans from the previous iteration [(X-1).Travel Plans] are reskimmed by the Router to produce a set of travel plans with travel times that are based on the travel times contained in the link delay file [X_RSK.Travel Plans]. When the Simulator is being used, these travel times are an average of the simulated travel times over a given time period, typically, 5 minutes.

Comparing the individual trip time directly from the performance file produced by the Traffic Simulator with the Router's calculated shortest path travel time is a comparison between a single simulation observation and a sum of average travel times for the set of links in the path. This difference in fidelity (simulated trip time compared with the sum of average simulated link travel times) is problematic as this comparison will likely return a non-zero result when the trip is on its shortest path. By definition based on the D.U.E. concept, the difference between the experienced travel time and shortest path travel time should be 0.

The approach used here to reconcile the difference in fidelity between experienced trip time from the Traffic Simulator and calculated shortest path time from the Router is to replace the simulated trip time for the gap calculation. This is accomplished by visiting each link the trip used, picking the average travel time for the 5 minute period the trip entered the link, and summing that for all the links in the trip path. This is sometimes called "re-skimming the path time". As this is done prior to calculating the trip gap, it enables a comparison between the current path travel time and shortest path travel time based on a set of link travel times at the same level of fidelity. Now the difference in travel time between a trip on its shortest path and the calculated shortest path travel time will always be zero.

While trips leaving a few minutes earlier or later in the time period may experience a different travel time, this approximation approach for calculating gap for traffic assignment solutions works consistently well in practice. Summarizing the link performance by using an average travel time over brief time intervals for both the shortest path and the experienced travel time provides sufficient fidelity to validate assignment models for typical planning applications. It is also important to note that this particular gap metric is not directly comparable to other gap metrics calculated using a different concept or underlying implementation.

Third, the RSK and AON plans are compared to produce a list of significantly different plans [X.PlanCompare]. The PlanCompare program also calculates the trip-based relative gap user equilibrium convergence statistic. The gap measure is calculated for each time increment used by the Router (typically, 5 to 15 minutes).

Fourth, the plans are merged [X.PlanMerge]. Unlike a traditional aggregate model, TRANSIMS is limited to one path for each traveler. For instance, if there are 10 trips between two points in a given time increment and there are two paths of nearly equal travel time, a traditional model will assign slightly more than half of the trips to one path and slightly less than half of the trips to the other path. TRANSIMS, on the other hand, will assign all of the trips to the shortest path. Merging the plans enables ***some*** of the trips to be shifted to the other path.

Finally, the link travel times and volumes from the previous iteration [(X-1).Link Delay] and the AON assignment [X_AON.Link Delay] are combined to produce a set of travel times and volumes [X.Link Delay] for the next iteration.

The primary implementation difference between the Dynamic User Equilibrium Convergence process and the Router or Microsimulator Stabilization processes is the use of the PlanCompare program. In the Router Stabilizer and Microsimulator Stabilizer processes, the PlanSelect program was used to select a subset of travelers to re-route. In the Dynamic User Equilibrium Convergence process, all of the travelers are re-routed using the latest travel times. The PlanCompare program then compares the new travel plans to the travel plans that were previously used. If the travel time for a given traveler is approximately the same in both plan files, the traveler could not improve their travel time by changing paths and would therefore satisfy the user equilibrium condition. If the times are different, the traveler has not achieved user equilibrium and would therefore qualify for selection.

6. Description of Relevant TRANSIMS Tools

EventSum - This program summarizes the differences between scheduled and actual start times, arrival times, and travel times. It also uses schedule constraints from a trip or activity file to adjust the start times in the plan file to enable the trip to reach the destination on-time given the simulated travel time.

IntControl – This program is primarily used to synthesize intersection controls. However, it also can be used to update the signal timing plans for selected locations and time periods based on turning movements or link delay data.

LinkDelay - This program merges, averages, and/or converts link delay files and smoothes the link delays between time increments.

LinkSum - This program summarizes link-based performance statistics, including comparisons between two link delay files.

PlanCompare - PlanCompare compares two plan files and selects the plans that have significantly different travel times or generalized costs. Then it generates convergence statistics and distribution charts by time-of-day.

PlanPrep - The selected plans are input to PlanPrep, which manipulates plan files by sorting, selecting or merging for future use by Simulator. PlanPrep generates distribution reports of path and travel time changes.

PlanSelect - This program selects plans based on traveler ID, time-of-day, activity location, parking lot, transit stop, transit route, V/C ratio, travel time ratio, coordinates, vehicle types, subarea polygon, and path node or link sequences to ultimately create a selection file that can be used as input to the Router in subsequent iterations. It can also use link delay information to select plans where the plan duration and the current travel time for the plan path are significantly different.

PlanSum - PlanSum summarizes the link demands generated by the Router, and applies volume-delay equations to estimate link travel times. The program produces link volumes, link delays, and turning movements by time-of-day. In addition, it produces zone or district trip tables and skim files by mode and time-of-day.

ProblemSelect - This program creates a list of household IDs (a selection file) based on trips with problems.

Router - The Router generates travel plans for household trips and tours that are connected by walk, drive, ride, transit, park-&-ride, kiss-&-ride, and bicycle modes. It builds travel plans from specified origins to specified destinations at specified times of day using a specified travel trip mode. Alternatively, the Router can selectively build paths from specified origins, to specified destinations, and at specified times of day, using specified modes. The Router can also be implemented using an incremental capacity restrained assignment algorithm. The output is an updated plan file based on new skims.

Simulator - The Simulator simulates the movement of vehicles in the network on a second-by-second basis as they follow their travel plans defined by the Router. The outputs of the Simulator are performance statistics, the ability to track individual travelers, and summaries of events.

PlanTrips - PlanTrips generates a trip file from a plan file and uses the time constraints from the original trip file and the trip duration from the plan files to update the trip start and end times in the new trip file and optionally the new plan file. The program adjusts the plan start and end times for each leg of a given traveler to remove overlapping travel legs and compress schedule gaps.

7. Control files in the demo example

7.1 Control Files

Two batch files are used to run this stabilization / user equilibrium sequence. The first performs an initial router run, and then two iterations of router stabilization.

```
path C:\Users\Public\Programs\Transims50\Win32\Bin
set TRANSIMS_CONFIG_FILE=config.txt

rem ---- Create Partitions and Route All Trips ----
RandomSelect.exe RandomSelect.ctl
Router.exe  1.Router.ctl

rem ---- Select Plans, Route the Selected Plans ----
PlanSelect.exe  1.PlanSelect.ctl
Router.exe  2.Router.ctl
PlanSelect.exe  2.PlanSelect.ctl
Router.exe  3.Router.ctl
pause
```

The second performs a router-based user equilibrium sequence.

```
path C:\Users\Public\Programs\Transims50\Win32\Bin
set TRANSIMS_CONFIG_FILE=config.txt
```

```
Router.exe  4.RouterAON.ctl
Router.exe 4.RouterRSK.ctl
LinkSum.exe     4.LinkSum.ctl

PlanCompare.exe  4.PlanCompare.ctl
LinkDelay.exe  4.LinkDelay.ctl

Router.exe  5.RouterAON.ctl
Router.exe 5.RouterRSK.ctl
LinkSum.exe     5.LinkSum.ctl
PlanCompare.exe  5.PlanCompare.ctl
LinkDelay.exe  5.LinkDelay.ctl

Router.exe  6.RouterAON.ctl
Router.exe 6.RouterRSK.ctl
LinkSum.exe     6.LinkSum.ctl
PlanCompare.exe  6.PlanCompare.ctl
LinkDelay.exe  6.LinkDelay.ctl

Router.exe  7.RouterAON.ctl
Router.exe 7.RouterRSK.ctl
LinkSum.exe     7.LinkSum.ctl
PlanCompare.exe  7.PlanCompare.ctl
LinkDelay.exe  7.LinkDelay.ctl

Router.exe  8.RouterAON.ctl
Router.exe 8.RouterRSK.ctl
LinkSum.exe     8.LinkSum.ctl
PlanCompare.exe  8.PlanCompare.ctl
LinkDelay.exe  8.LinkDelay.ctl

Router.exe  9.RouterAON.ctl
Router.exe 9.RouterRSK.ctl
LinkSum.exe     9.LinkSum.ctl
PlanCompare.exe  9.PlanCompare.ctl
LinkDelay.exe  9.LinkDelay.ctl

Router.exe  10.RouterAON.ctl
Router.exe 10.RouterRSK.ctl
LinkSum.exe     10.LinkSum.ctl
PlanCompare.exe  10.PlanCompare.ctl

pause
```

7.2 Initial Router Run

See Figure 2. The RandomSelect control file is as follows. It is optional, and divides the trips into partitions, to enable parallel processing by the Router and other programs.

```
TITLE                      RandomSelect Test
TRIP_FILE                  demand/trip.txt
NEW_SELECTION_FILE         demand/select.txt
NUMBER_OF_PARTITIONS       2
```

The Router control file is as follows.

```
TITLE                      Router Test
SELECTION_FILE             demand/select.txt
TRIP_FILE                     demand/trip.txt
VEHICLE_FILE                  demand/vehicle.txt
VEHICLE_TYPE_FILE                inputs/vehicle_type.txt

NODE_FILE                  network/node.txt
LINK_FILE                  network/link.txt
CONNECTION_FILE            network/connection.txt
POCKET_FILE                   network/pocket.txt
PARKING_FILE                  network/parking.txt
LOCATION_FILE                 network/location2.txt

NEW_PLAN_FILE              demand/1.plans.*
NEW_PROBLEM_FILE           demand/router_problems.txt
NEW_LINK_DELAY_FILE           results/1.linkdelay.txt

UPDATE_FLOW_RATES             YES
CLEAR_INPUT_FLOW_RATES           YES
UPDATE_TRAVEL_TIMES            YES
LINK_DELAY_UPDATE_RATE           -1
```

7.3 Router Stabilization

See Figure 3. The first PlanSelect control file is as follows. In this case, V/C ratios are used to select plans.

```
TITLE                  Select Travelers to Re-Route in 1.Router

PLAN_FILE                        demand/1.plans.*
LINK_DELAY_FILE                  results/1.linkdelay.txt
```

```
NEW_SELECTION_FILE               demand/1.Select.*

NODE_FILE                  network/node.txt
LINK_FILE                  network/link.txt
CONNECTION_FILE            network/connection.txt
POCKET_FILE             network/pocket.txt
PARKING_FILE            network/parking.txt
LOCATION_FILE           network/location2.txt

SELECT_VC_RATIOS                1.1        //--Baseline 1.1
RANDOM_NUMBER_SEED               234
```

The first Router control file is as follows. Major differences between this file and the file for the initial run include the following:

- The selection file indicates that only a few trips are to be routed
- A link delay file is used as inputs, so that routing is based on congested travel times from the initial run.

```
TITLE                      Router Test
SELECTION_FILE             demand/1.Select.*
TRIP_FILE                     demand/trip.txt
VEHICLE_FILE                  demand/vehicle.txt
VEHICLE_TYPE_FILE             inputs/vehicle_type.txt

NODE_FILE                  network/node.txt
LINK_FILE                  network/link.txt
CONNECTION_FILE            network/connection.txt
POCKET_FILE                   network/pocket.txt
PARKING_FILE                  network/parking.txt
LOCATION_FILE                 network/location2.txt

LINK_DELAY_FILE                   results/1.linkdelay.txt
PLAN_FILE                             demand/1.plans.*

NEW_PLAN_FILE              demand/2.plans.*
NEW_PROBLEM_FILE           demand/router_problems.txt
NEW_LINK_DELAY_FILE           results/2.linkdelay.txt

UPDATE_FLOW_RATES             YES
UPDATE_TRAVEL_TIMES        YES
CLEAR_INPUT_FLOW_RATES           YES
```

For subsequent iterations, the input and filenames are changed, with numbers (e.g., 1.linkdelay.txt) incrementing by one with each iteration. For example, in the second iteration, the following parameters are changed in the router control file:

```
SELECTION_FILE             demand/2.Select.*
LINK_DELAY_FILE                   results/2.linkdelay.txt
PLAN_FILE                             demand/2.plans.*
NEW_PLAN_FILE              demand/3.plans.*
NEW_LINK_DELAY_FILE           results/3.linkdelay.txt
```

7.4 Macroscopic User Equilibrium

The Router control file, for iteration 4, is as follows. Like the initial router run, it makes use of all trips. However, it also uses the link delay file from the previous run (3.linkdelay.txt) to route using congested travel times.

```
TITLE                      Router AON Assignment

SELECTION_FILE             demand/select.txt
TRIP_FILE                     demand/trip.txt
VEHICLE_FILE                  demand/vehicle.txt
VEHICLE_TYPE_FILE                inputs/vehicle_type.txt

NODE_FILE                  network/node.txt
LINK_FILE                  network/link.txt
CONNECTION_FILE            network/connection.txt
POCKET_FILE                   network/pocket.txt
PARKING_FILE                  network/parking.txt
LOCATION_FILE                 network/location2.txt

LINK_DELAY_FILE                  results/3.linkdelay.txt
NEW_PLAN_FILE              demand/4AON.plans.*
NEW_PROBLEM_FILE           demand/router_problems.txt
NEW_LINK_DELAY_FILE           results/4AON.linkdelay.txt

UPDATE_FLOW_RATES             YES
CLEAR_INPUT_FLOW_RATES           FALSE
UPDATE_TRAVEL_TIMES            YES
LINK_DELAY_UPDATE_RATE           -1
```

For subsequent iterations, filenames change in this file, similar to the changes for router stabilization.

The linkdelay control file, for iteration 4, is as follows.

```
TITLE                          Link Delay Averaging for TestNet

NODE_FILE                               network/node.txt
LINK_FILE                               network/link.txt
CONNECTION_FILE                      network/connection.txt
PARKING_FILE                            network/parking.txt
LOCATION_FILE                           network/location2.txt
POCKET_FILE                             network/pocket_lane.txt

LINK_DELAY_FILE               results/4AON.linkdelay.txt
MERGE_LINK_DELAY_FILE          results/3.linkdelay.txt
NEW_LINK_DELAY_FILE            results/4.linkdelay.txt

MERGE_WEIGHTING_FACTOR               1
PROCESSING_METHOD               WEIGHTED_AVERAGE
```

For this file, the MERGE_WEIGHTING_FACTOR is set equal to 1, so that the two input files, 4AON.linkdelay.txt and 3.linkdelay.txt, are given equal weight. Subsequent iterations will implement a method of successive averages (MSA) scheme, where the MERGE_WEIGHTING_FACTOR is incremented by one for each iteration. This will give increasing weight to the merge link delay file for iterations subsequent to iteration 4. The following parameters are changed in subsequent iterations.

	Iteration			
Parameter	4	5		n
LINK_DELAY_FILE	4AON….txt	5AON….txt	…	nAON.linkdelay.txt
MERGE_LINK_DELAY_FILE	3.linkdelay.txt	4.linkdelay.txt	…	(n-1).linkdelay.txt
NEW_LINK_DELAY_FILE	4.linkdelay.txt	5.linkdelay.txt	…	n.linkdelay.txt
MERGE_WEIGHTING_FACTOR	1	2	…	(n-3)

7.5 Link-Based Relative Gap

The router control file was presented in the previous section. The link sum control file is as follows.

```
TITLE                     LinkSum Default Control Keys
#---- System File Keys ----
NODE_FILE                  network/node.txt
LINK_FILE                  network/link.txt
CONNECTION_FILE            network/connection.txt
POCKET_FILE                   network/pocket.txt
PARKING_FILE                  network/parking.txt
LOCATION_FILE                 network/location2.txt
PERFORMANCE_FILE           results/4AON.linkdelay.txt
COMPARE_PERFORMANCE_FILE   results/3.linkdelay.txt

#---- Data Service Keys ----
```

```
DAILY_WRAP_FLAG                         FALSE
SUMMARY_TIME_RANGES                     7.0..9.0 hours
SUMMARY_TIME_INCREMENT                  15 minutes

#---- LinkSum Control Keys ----
LINKSUM_REPORT_1      RELATIVE_GAP_REPORT
```

7.6 Trip-Based Relative Gap

The all or nothing (AON) router control file was presented in Section 7.4. The reskim (RSK) router control file is as follows. This router control file has an input plan file (3.plans.*) but no input trip file. It updates the travel times on a set of plans using the most recent available linkdelays.

```
TITLE                      Router Test

SELECTION_FILE             demand/select.txt
VEHICLE_FILE                  demand/vehicle.txt
VEHICLE_TYPE_FILE                inputs/vehicle_type.txt

NODE_FILE                  network/node.txt
LINK_FILE                  network/link.txt
CONNECTION_FILE            network/connection.txt
POCKET_FILE                   network/pocket.txt
PARKING_FILE                  network/parking.txt
LOCATION_FILE                 network/location2.txt

LINK_DELAY_FILE                  results/3.linkdelay.txt
PLAN_FILE       demand/3.plans.*
NEW_PLAN_FILE              demand/4RSK.plans.*
NEW_PROBLEM_FILE           demand/router_problems.txt

UPDATE_FLOW_RATES             NO
CLEAR_INPUT_FLOW_RATES           FALSE
UPDATE_TRAVEL_TIMES             YES
LINK_DELAY_UPDATE_RATE           -1
UPDATE_PLAN_RECORDS          TRUE
```

The PlanCompare control file appears below. In addition to producing a trip gap report, this control file also produces a list of plans that are different, to be used in merging plans.

```
TITLE                      PlanCompare Test

PLAN_FILE       demand/4AON.plans.*
COMPARE_PLAN_FILE          demand/4RSK.plans.*
NEW_PLAN_FILE              demand/4DIF.plans.*

PERCENT_TIME_DIFFERENCE          10
MINIMUM_TIME_DIFFERENCE          1
MAXIMUM_TIME_DIFFERENCE          10
SELECTION_PERCENTAGE       50
MAXIMUM_PERCENT_SELECTED   10
SUMMARY_TIME_RANGES        ALL
PLANCOMPARE_REPORT_1 TRIP_TIME_GAP_REPORT
PLANCOMPARE_REPORT_2 TRIP_COST_GAP_REPORT
```