TRANSIMS Training Course at TRACC
Transportation Research and Analysis Computing Center

Part 5

Routing on the Street and Transit Networks in TRANSIMS

Dr.-Ing. Hubert Ley
Michael Hope
Transportation Research and Analysis Computing Center

Last Updated: December 9, 2009

Unit **5**

U.S. DEPARTMENT OF
ENERGY

Contents

- General overview over the TRANSIMS Router
- Trip Tables and Activity Tables (Primary Router Input)
- The Modified Dijkstra's Algorithm
- Plan Files (Primary Router Output)
- Network Layers and Internal Representation of the Network
- Travel Modes in the Router
- Travel Legs in Router Plan Files
- Label Constraints Based on Travel Modes
- Trip-based and Activity-based Routing
- Internal Link Delay Updates to Improve Initial Plans
- Initial and Subsequent Iterations, Plan Selection
- Partitioning on High Performance Computing Environments

The TRANSIMS Router

- The Route Planner module generates routes for each individual travel plan from a trip file
- The router can process both trips resulting from activities as well as simple trip tables, or both at the same time (e.g. trips for itinerant travelers plus trips from activity-based models)
- Trips can occur between any two activity locations (parking lot, transit stop, etc…)
- Starting time is supplied as well as an estimate on the arrival time
- Router builds a step-by-step travel plan that will take the person from their origin location to their destination location
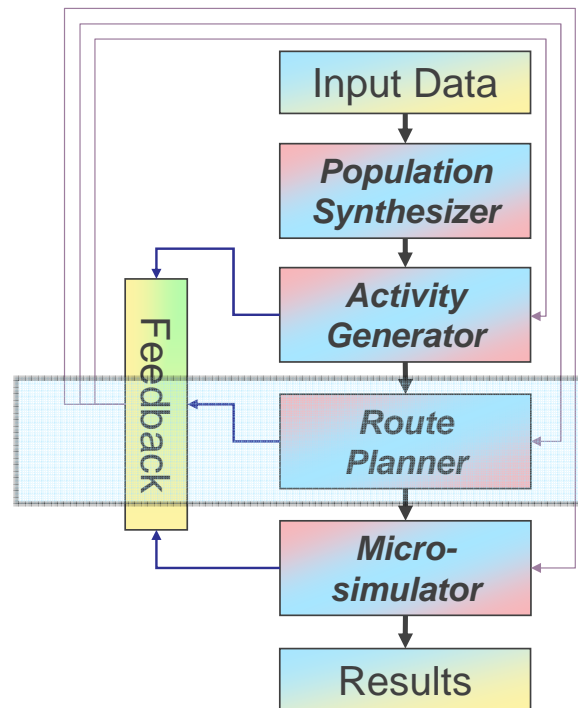
3

Router Demonstration

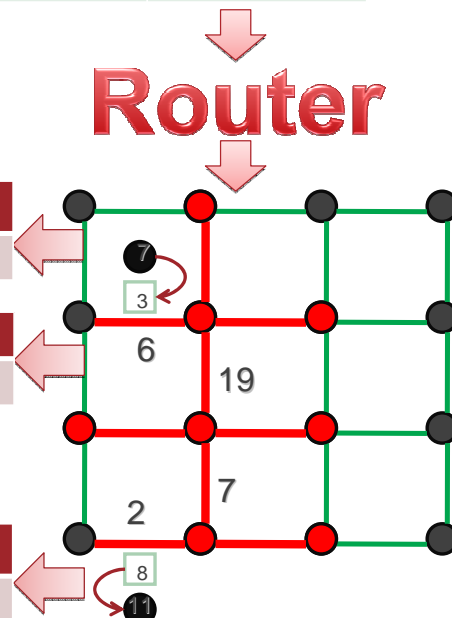| Trip #	Orig_Loc	Des_Loc	Start_Time	Approx_End
1	7	11	8:03	8:08

Leg #	Orig	Orig_Type	Dest	Dest_Type	Start_Time	End_Time
1	7	Activity	3	Parking	8:03:00	8:03:30

Leg #	Orig	Orig_Type	Dest	Dest_Type	Start_Time	End_Time
2	3	Parking	8	Parking	8:03:30	8:07:45

Path:			
6	19	7	2

Leg #	Orig	Orig_Type	Dest	Dest_Type	Start_Time	End_Time
3	8	Parking	11	Activity	8:07:45	8:08:15

4

Primary Router Inputs (Example Trip Table)

- The example below is based on trip-based modeling
- Yellow columns are the subject of this slide
- When activity tables are processed, similar trips information is derived internally from the given activity pattern
- Arrival time in this case is estimated by ConvertTrips (more in the session on trip-based modeling)
- Activities and trips belong to individual persons in individual households
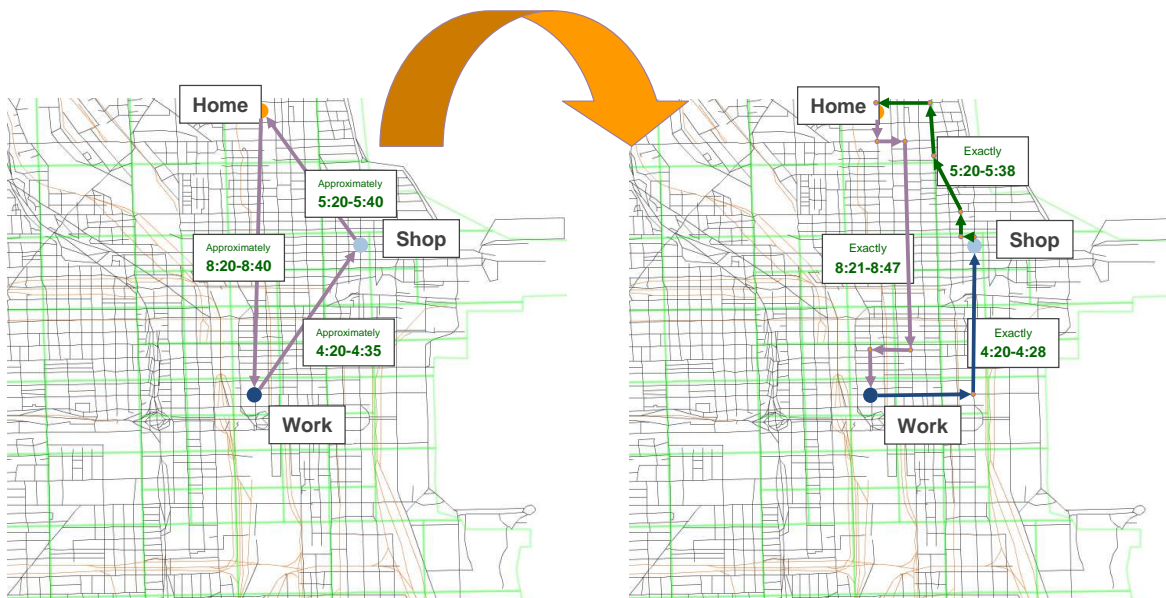- Trips start and end at activity locations

HHOLD	PERSON	TRIP	PURPOSE	MODE	VEHICLE	START	ORIGIN	ARRIVE	DESTINATION	CONSTRAINT
1	1	1	1	2	1	20352	937	22317	7211	1
2	1	1	1	2	2	27336	3133	29282	1920	1
3	1	1	1	2	3	28914	1879	30659	769	1
4	1	1	1	2	4	24796	1899	26319	781	1
5	1	1	1	2	5	23843	1863	25507	879	1

The TRANSIMS Router

- The Route Planner computes a time-dependent shortest path for each travel-leg using a modified version of Dijkstra's algorithm

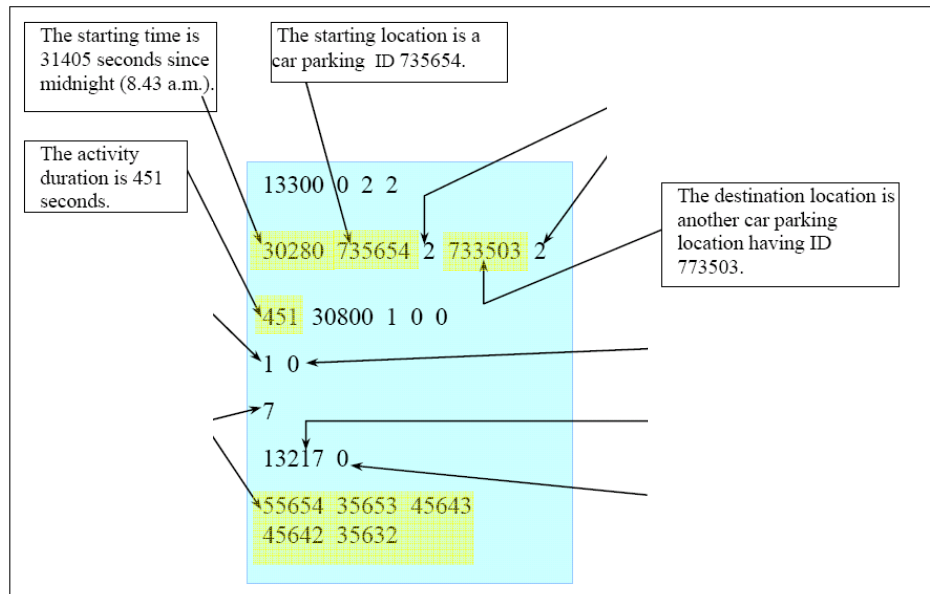## Primary Router Outputs (Plan File)

The starting time is 31405 seconds since midnight (8.43 a.m.).

The starting location is a car parking ID 735654.

The activity duration is 451 seconds.

The destination location is another car parking location having ID 773503.

13300 0 2 2

30280 735654 2 733503 2

451 30800 1 0 0

1 0

7

13217 0

55654 35653 45643
45642 35632

Figure-5.18: Example of an output for the "car" mode
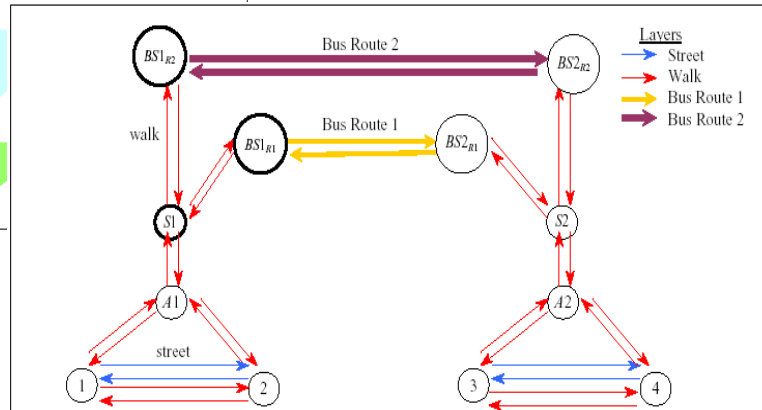

Network Layers

- The network is represented in form of transportation mode layers
- Each transit route is represented by its own layer
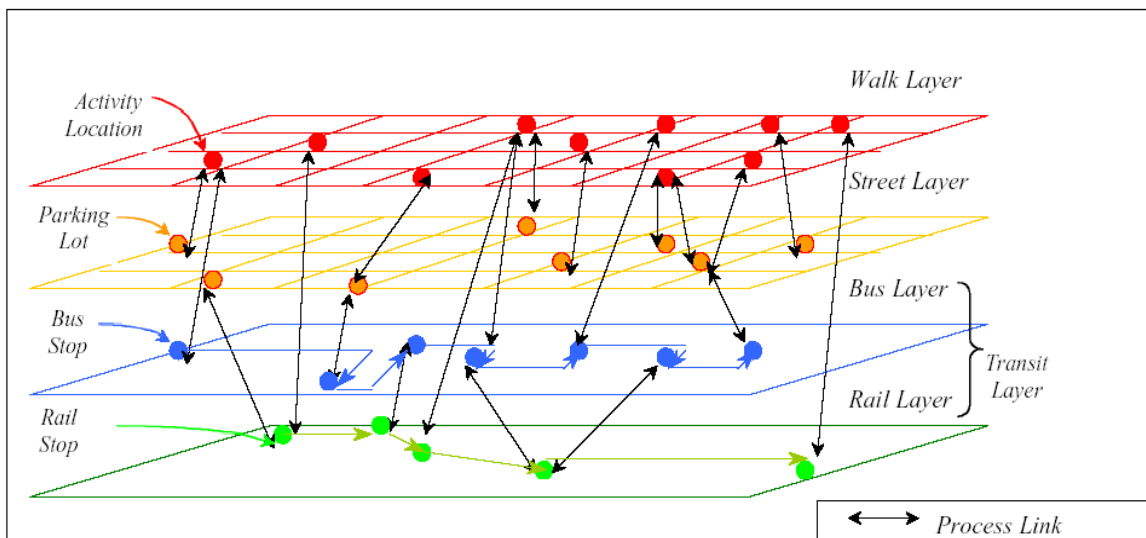
Another Network View
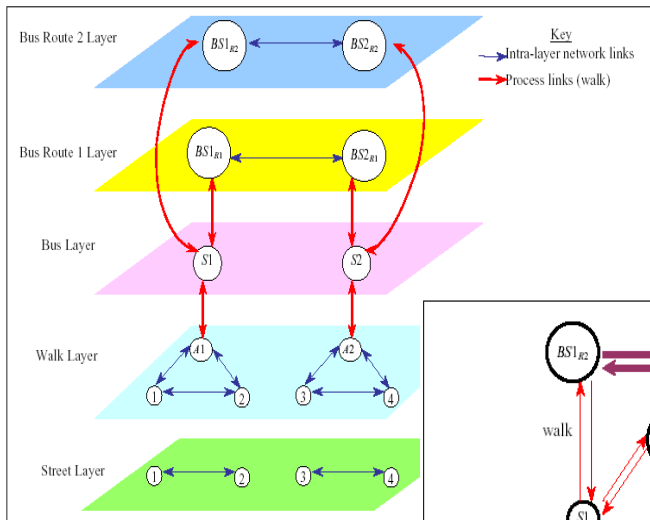
- Example network of
  - Four network nodes
  - Two network links (not interconnected)
  - Two transit stops
  - Two lines utilizing these transit stops
  - Process links as required

Travel Modes

- Trip table contains a key called Mode which specifies the types of transportation methods available to the person to get from place to place

- Important Modes Include:
  - 1: Only Walking
  - 2: Walking or Car
  - 3: Walking or Transit
  - 5 & 6: Walking or Transit or Auto

- Transims keeps track of vehicles. If a mode is specified that uses a car, a vehicle identifier must also be specified.
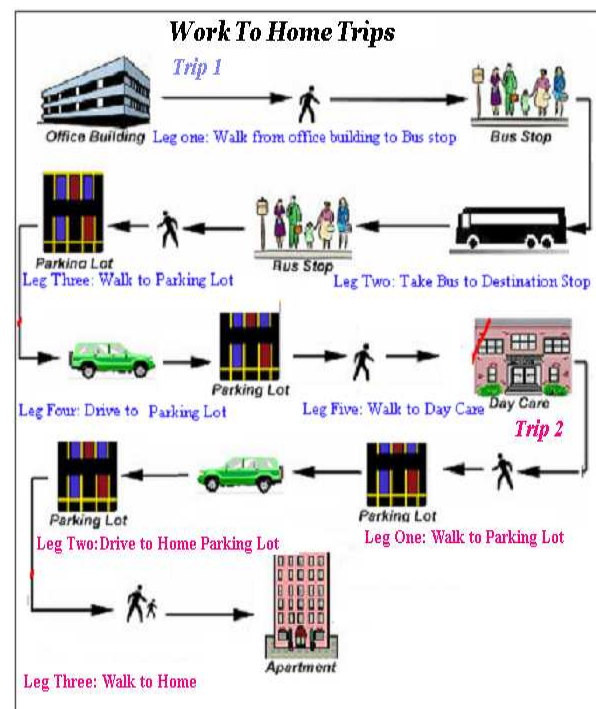
Router Mode Inputs (Trip Table)

- The is a similar trip table example as shown a few slides ago
- Yellow columns with regards to vehicle and travel mode are the subject of this slide
- One or more individual vehicles are assigned to each household
- Shared trips are processed together for all household members
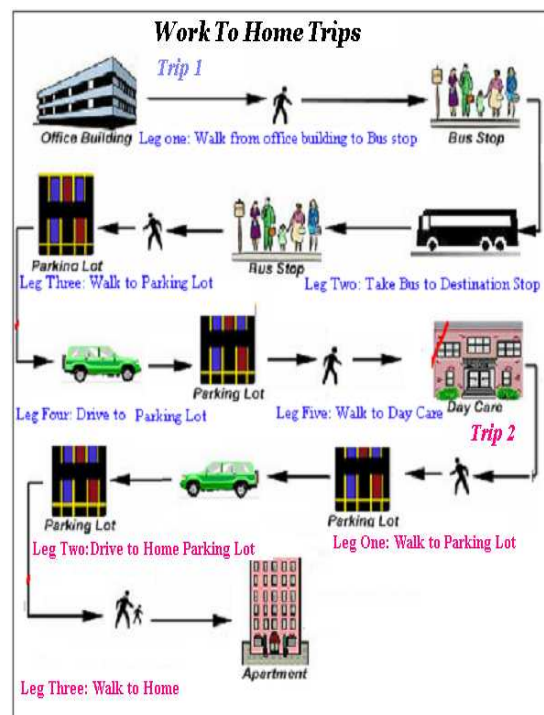- Travel mode codes identify the mode of transportation (Car, bus, train, walking)

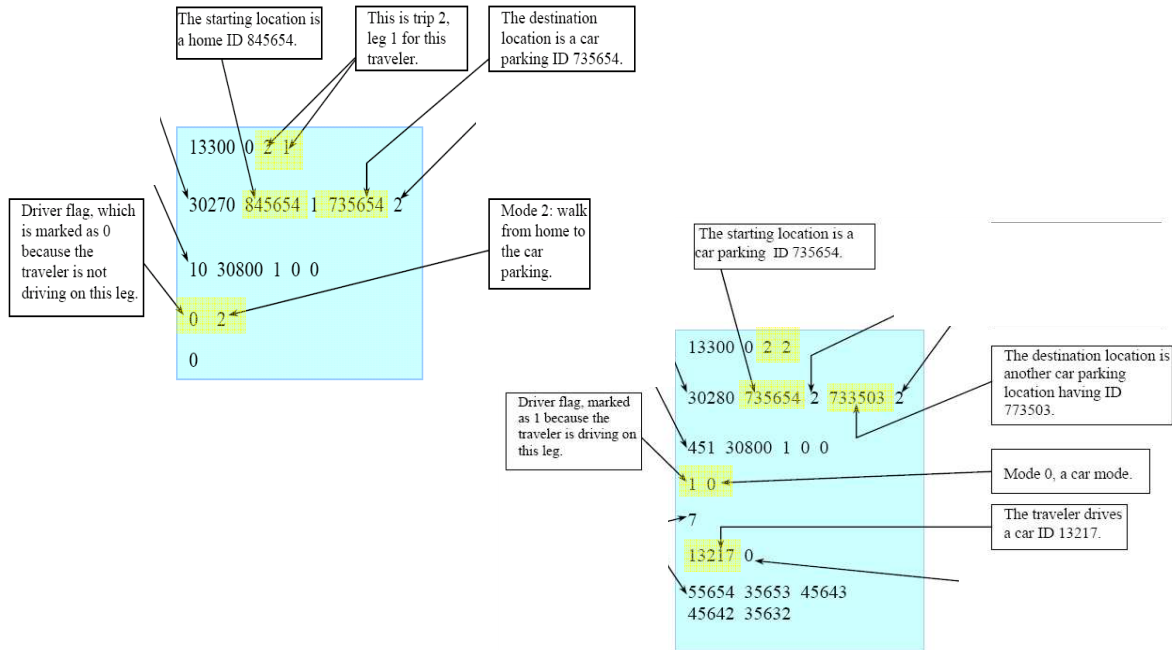| HHOLD	PERSON	TRIP	PURPOSE	MODE	VEHICLE	START	ORIGIN	ARRIVE	DESTINATION	CONSTRAINT
1	1	1	1	2	1	20352	937	22317	7211	1
2	1	1	1	3	0	27336	3133	29282	1920	1
3	1	1	1	1	0	28914	1879	30659	769	1
4	1	1	1	2	2	24796	1899	26319	781	1
5	1	1	1	5	3	23843	1863	25507	879	1

Trip Legs and Mode Labels

- Using the Mode information, Router divides the trip up into legs
- Each leg represents a chronological step in the route which only uses a certain travel type
- Plan file labels travel types as follows:
  - Drive (D) = 0
  - Transit (T) = 1
  - Walk (W) = 2
- Sequence of legs are specific to Mode used:
  - (Walk): W
  - (Drive): W->D->W
  - (Transit): W->T->W
  - (Park+Ride): W->D->W->T->W
  - (Park+Ride): W->T->W->D->W

Work To Home Trips

Trip 1

Office Building Leg one: Walk from office building to Bus stop   Bus Stop

Parking Lot
Leg Three: Walk to Parking Lot   Bus Stop   Leg Two: Take Bus to Destination Stop

Leg Four: Drive to Parking Lot   Parking Lot   Leg Five: Walk to Day Care   Day Care

Trip 2

Parking Lot
Leg Two: Drive to Home Parking Lot   Parking Lot   Leg One: Walk to Parking Lot

Apartment
Leg Three: Walk to Home

Two Consecutive Trip Legs

The starting location is a home ID 845654.

This is trip 2, leg 1 for this traveler.

The destination location is a car parking ID 735654.

13300 0 2 1

30270 845654 1 735654 2

Driver flag, which is marked as 0 because the traveler is not driving on this leg.

10 30800 1 0 0

0 2

0

Mode 2: walk from home to the car parking.

The starting location is a car parking ID 735654.

13300 0 2 2

30280 735654 2 733503 2

Driver flag, marked as 1 because the traveler is driving on this leg.

451 30800 1 0 0

1 0

7

13217 0

55654 35653 45643
45642 35632

The destination location is another car parking location having ID 773503.
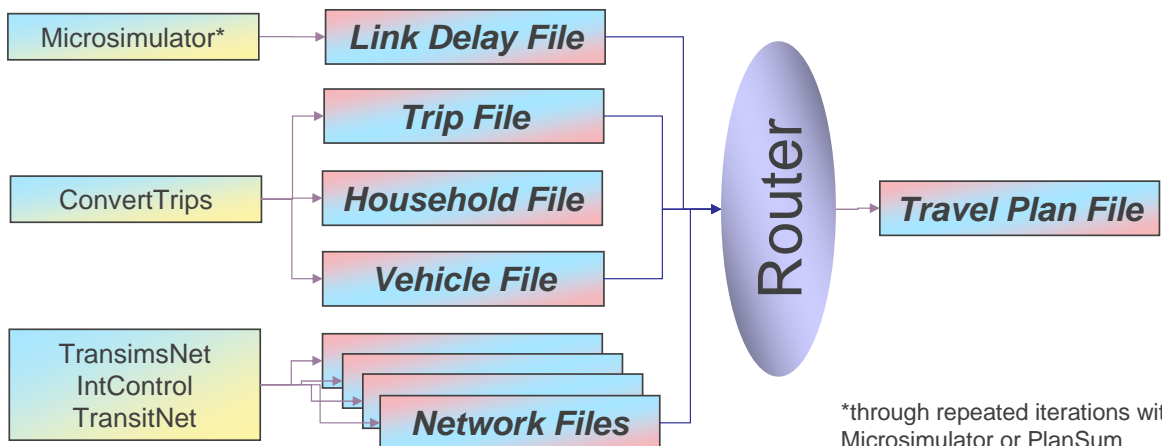
Mode 0, a car mode.

The traveler drives a car ID 13217.

13

## The TRANSIMS Router for <u>Trip-Based</u> Models

- The router uses a <u>trip file generated by ConvertTrips</u> as its main input. Additional inputs are link delays (iteratively obtained from the microsimulator or PlanSum), the road and transit networks, as well as the household and vehicle files.

Microsimulator* → *Link Delay File*

Trip File

ConvertTrips → *Household File*

Vehicle File

TransimsNet
IntControl
TransitNet → *Network Files*

Router → *Travel Plan File*

*through repeated iterations with Microsimulator or PlanSum

14

The TRANSIMS Router for <u>Activity-Based</u> Models

- For a more realistic simulation, the router can use an activity-based method
- The router uses an <u>activity file generated by the activity generator</u> as its main input. Additional inputs are link delays (iteratively obtained from the microsimulator or PlanSum), the road and transit networks, as well as the household and vehicle files.

*through repeated iterations with Microsimulator or PlanSum
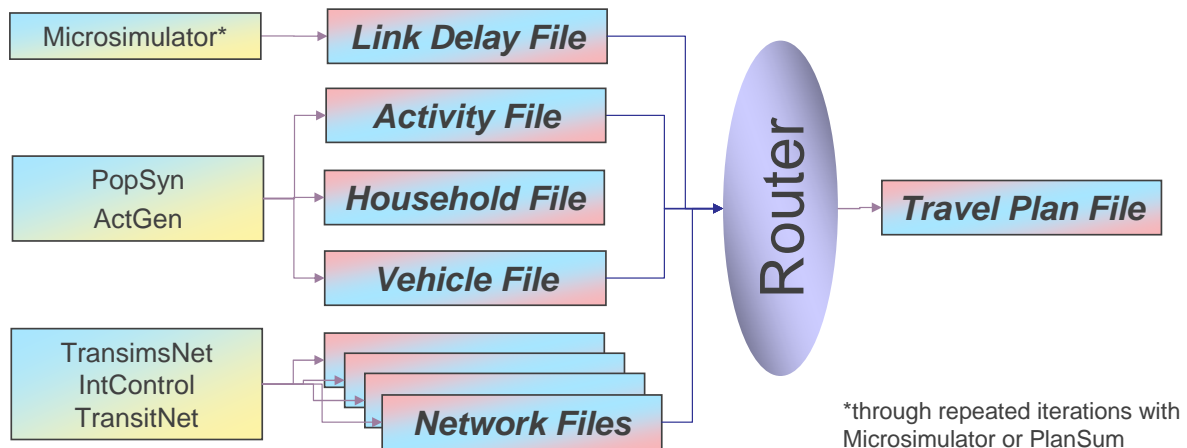
15

The TRANSIMS Router

- When the router reads an activity or trip file, it runs the plan builder routines for each trip independently. This means that there are no time-based interactions between plans. Everyone "has the road to themselves."
- Exception: To allow members of a household to share rides, members of a household must be routed as a group (in successive order))
- Due to this independent nature of routing, the router needs to use a few tools to make realistic plans.
  - Link Delays: decrease the speed along certain roads, thus increasing calculated travel time and emulating congestion. The router uses a shortest path algorithm, so it will route traffic around these congested areas whenever possible.
  - BPR+ Formulas: During the first iteration, when link delays are not yet available, the router can optionally calculate link delays on the fly based on BPR+ formulas, creating some level of dependencies between trips
  - Random Noise Functions: (± 5% or similar) may be added to link travel times to give a more comprehensive routing picture and ultimately decrease the number of iterations necessary

16

# Typical Application of the TRANSIMS Router

- BPR+ traffic assignment functions during the first iteration.
  - The router is interrupted every few thousand trips and the number of trips through each link at every time interval is evaluated using BPR+ functions
  - The resulting link delays for all links and all time intervals are updated in the link delay matrix, and operation continues
  - The resulting link delays are good enough to provide a starting point for further iterations (although the algorithm cannot deliver an actual result)

- Alternatively, incremental loading is used to route trips in an iterative loop with PlanSum
  - Load an additional random 5% of trips during each iteration, and run PlanSum after the router is done to determine link delays for the next iteration

---

Typical Application of the Router

- For subsequent iterations (after the initial loading using either methodology outlined on the previous slide), only subsets of travelers are being rerouted
  - Rerouting of travelers having already close to optimal routes is inefficient and does not contribute to the equilibration
  - Rerouting of all travelers will lead to oscillations in the equilibration process and not lead to results (see plot above – more in the session on Feedback and Equilibration)

- The selection of trips to reroute can be accomplished by using tools such as PlanSelect to set the specific criteria for selection
  - For example, large difference in estimated and actual arrival times
  - Trips that go through specific nodes and links
  - Trips at specific times of the day, and more
  - Random selections of the above subsets to limit the number of households being rerouted
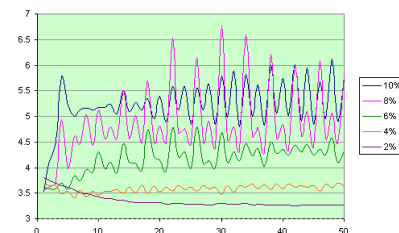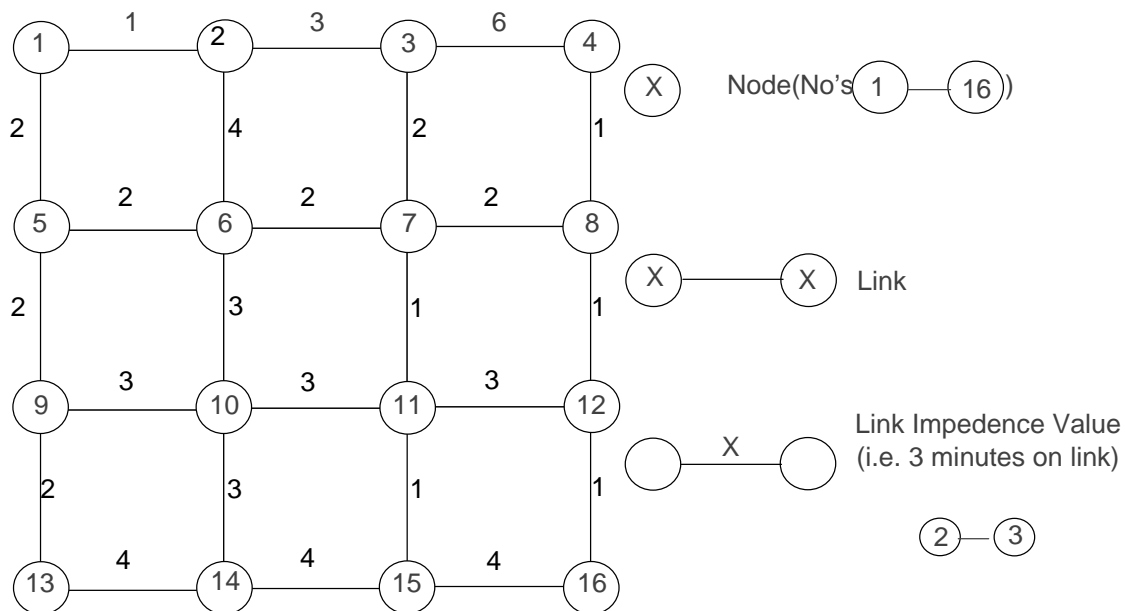
Algorithms

- The algorithm adopted is a variant of Dijkstra's procedure for finding shortest paths, modified to accommodate time-dependent link delays and label sequence constraints
  - The underlying problem is referred to as the "Time-Dependent Label-Constrained Shortest Path Problem (TDLSP)"

- Based on the Moore algorithm
  - Originated from work to solve the route selection of the telephone systems for direct dialing of long distance telephone calls
  - The first and most widely used algorithm is referred to as the Moore algorithm
  - This algorithm does not require all possible routes between an origin and destination to be individually investigated to find the shortest route

19

Simple Network Example

20

Simple Network Example - Algorithm

- Start at node 1 and find shortest path to node 16
- From the first node, examine travel times to all connecting nodes
- Continue from each node
- When reaching a node from multiple links, only keep the shortest link
- The algorithm doesn't only find the shortest path from 1 to 16, but actually the shortest path from 1 to any other node
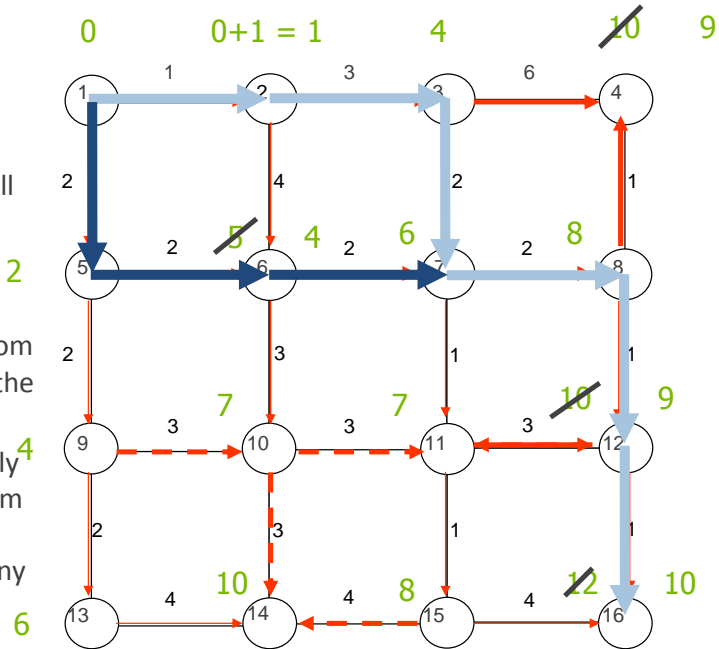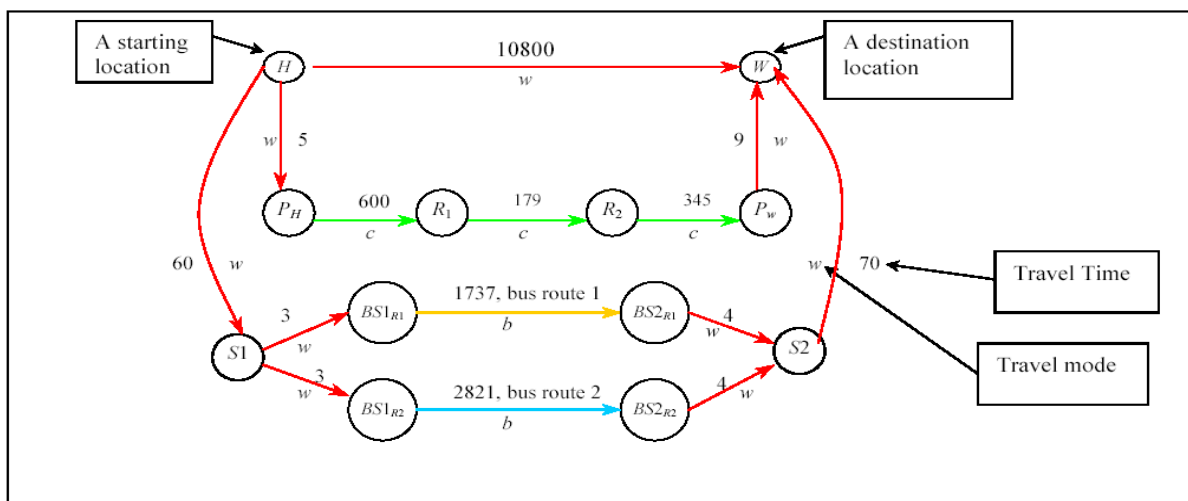

Visual Representation of Tree Building

Time-Independent Label Constrained Shortest Path (TILSP)

| Person ID	Trip Number	Starting Location	Destination Location	Starting Time (seconds since midnight)	Maximum Travel Time (second)	Mode String
13300	1	845654 (Home)	833503 (Work)	28800 (8.00 a.m.)	3000 *	wcw or wbw

Time-Dependent Label Constrained Shortest Path (TDLSP)

- The TRANSIMS router uses a time-dependent algorithm in addition to the algorithms outlined on the previous slides

- The travel time across the links is now a function of time

  – For the initial routing, this comes from free speeds

  – For iterative routing, this comes from the microsimulator
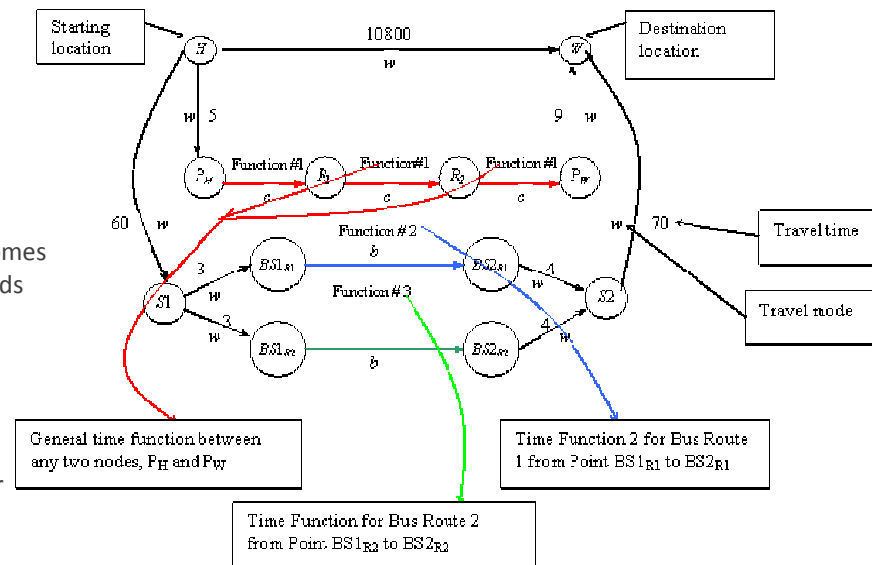
23

Output from the Router

- The output from the router are plans and problems
  – The plans are the major result from routing and contain the exact list of nodes or links that the traveler goes through, including the start and end time
  – Problems occur for a number of reasons (exceedingly large travel times, unreasonable connections, impassable network objects) and are listed so that they can be addressed in future iterations
- The trip plans contain timing information as available to the router
  – The timing information may be determined to be inadequate by the microsimulator, which results in an update of the actual link speeds at that time of day for the next iteration of the router
- The microsimulator executes the trip plans on another representation of the network, thus taking into consideration more detail (traffic signals, number of lanes, congestion, etc.) than the router
  – Vehicles and individuals that are unable to follow the specific route as determined by the router cannot be dynamically rerouted and have to be removed from the microsimulator, resulting in "problems"

24

New in Version 4 of the Router

- Partitioning: Due to the fact that routing is rather parallel in nature, the total number of trips can be partitioned into smaller subsets by household
  - Different partitions can be routed separately on different computers (but without load balancing)
  - The partitioning carries through consistently to other related TRANSIMS tools such as PlanPrep, PlanSelect, etc.
  - The partitioning logic will be maintained on a file compatibility level in the parallel router, but load balancing will allow for better performance and easier operation
- Link delays
  - During the first iteration, estimated link delays can be determined within the router based on the trips processed so far, resulting in a reasonable starting point for equilibration
  - Link delays calculated by the microsimulator support separate queues for left and right turns in addition to the traditional link delays; these queues are considered in the router algorithm as well

25

Credits and Acknowledgements

- GIS visualization materials were mostly developed at Argonne based on the TRANSIMS tools developed by AECOM for USDOT
- Chicago road and transit network data used in some of the examples was provided by the Chicago Metropolitan Agency for Planning
- USDOT provided the funding for the development of these training materials
- USDOT provided the funding for the TRACC computing center and the resources necessary to perform these training session
- Some figures have been developed for USDOT by Prof. Antoine Hobeika, Virginia Polytechnic Institute, Civil and Environmental Engineering
- The presentation is loosely based on materials provided by USDOT at a training course in November 2006

26