Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

UChicago Argonne LLC

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

TRANSIMS Training Course at TRACC

Transportation Research and Analysis Computing Center

Part 6

Microsimulation Using Cellular Automata on the Street Network

Dr.-Ing. Hubert Ley

Transportation Research and Analysis Computing Center

David Roden

AECOM, DMJM HARRIS

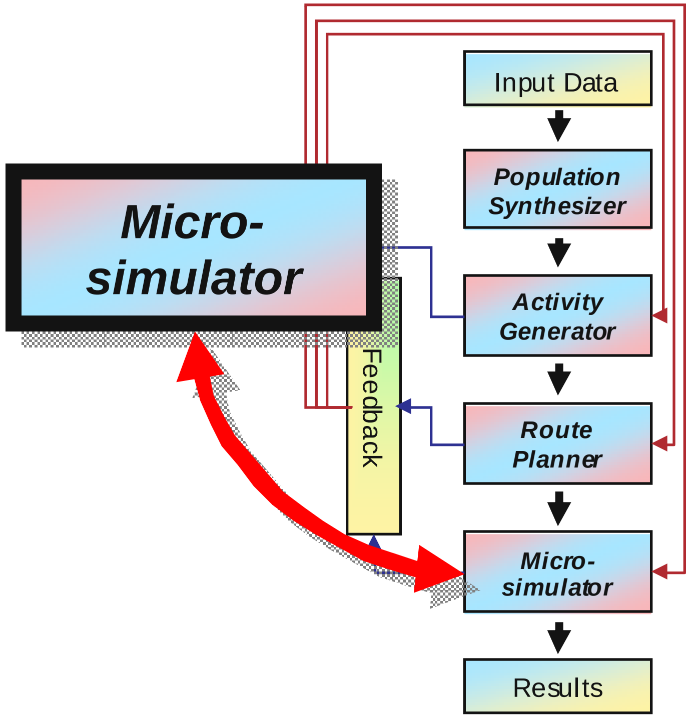
Vadim Sokolov

Transportation Research and Analysis Computing Center

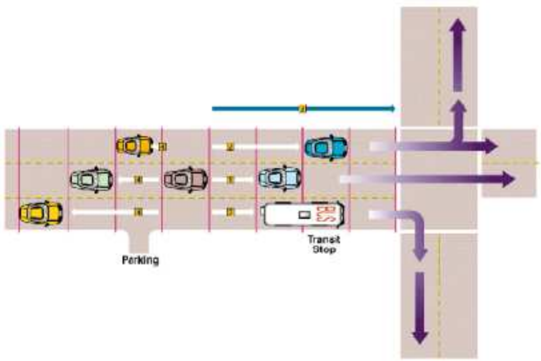
Last Updated: June 12, 2008

Contents

Purpose and Functions of the Microsimulator



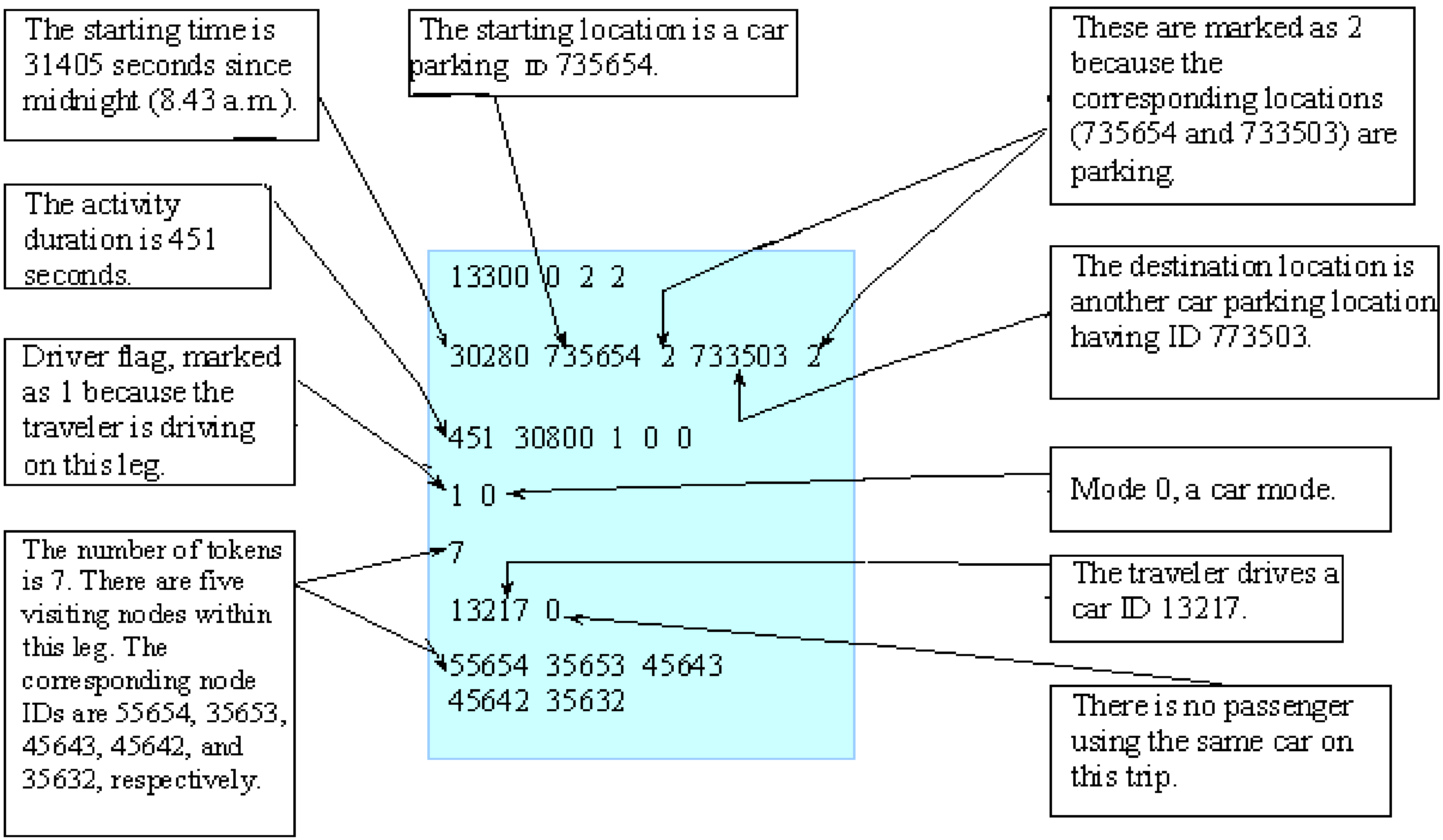
- Simulate the second-by-second movements of vehicles through the network.
- Follow exactly the travel plans determined by the router for each individual traveler
- Generate performance statistics, track individual travelers, and summarize events.



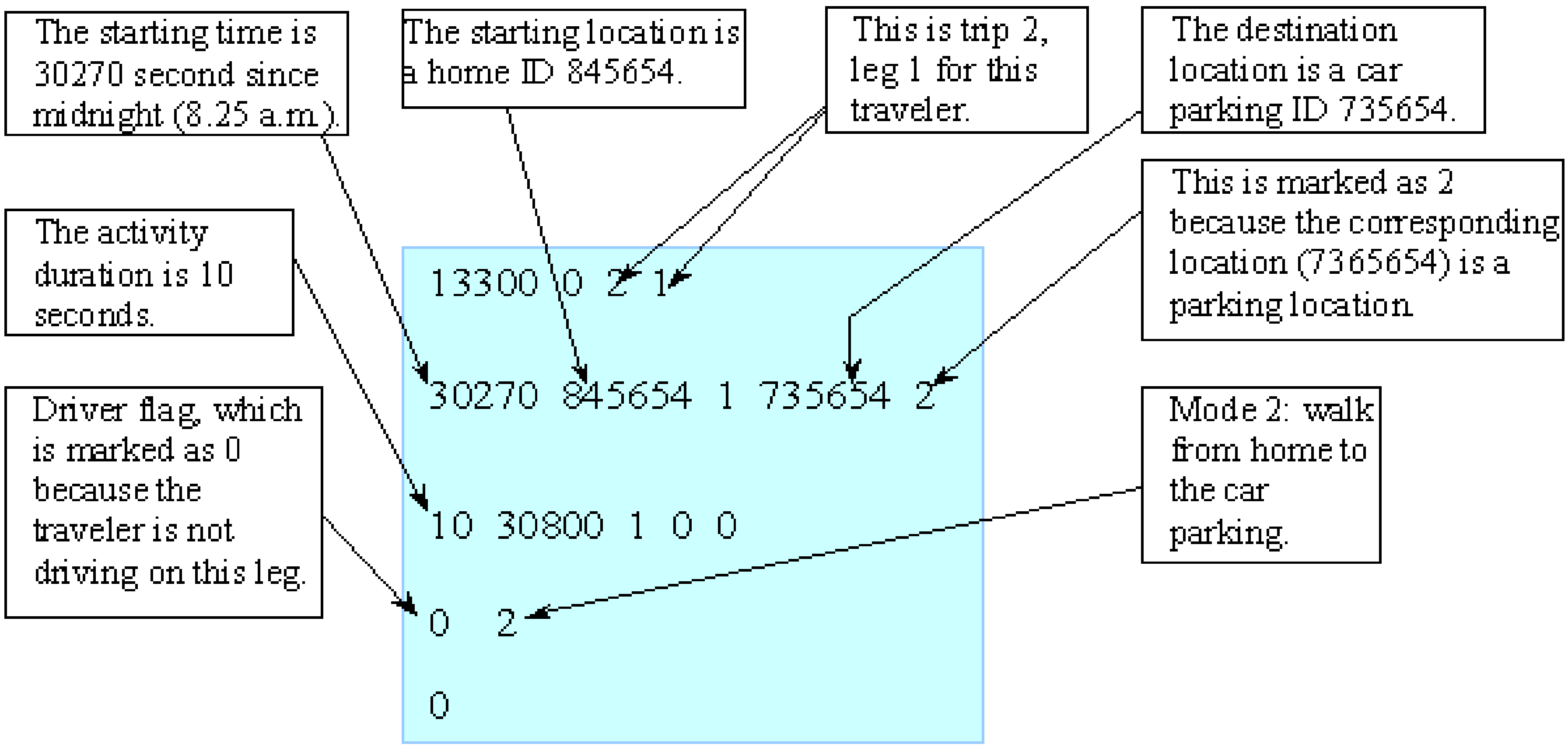
Input Required by the Microsimulator

- Network Data
  - Required files include Node, Link, Pocket Lane, Lane Connectivity, Parking, Activity Location, and Process Link
  - Optional files include Lane Use, Turn Prohibitions, Transit Stop, Transit Route, Transit Schedule, Transit Driver, Unsignalized Node, Signalized Node, Timing Plan, Phasing Plan, Detector, Signal Coordinator, and Boundary Speeds
- Time-sorted Plan File
  - Generated by the Router
- Vehicle File and Vehicle Type File
  - Generated by the population PopSyn or ConvertTrips
- Execution parameters, specified in control file

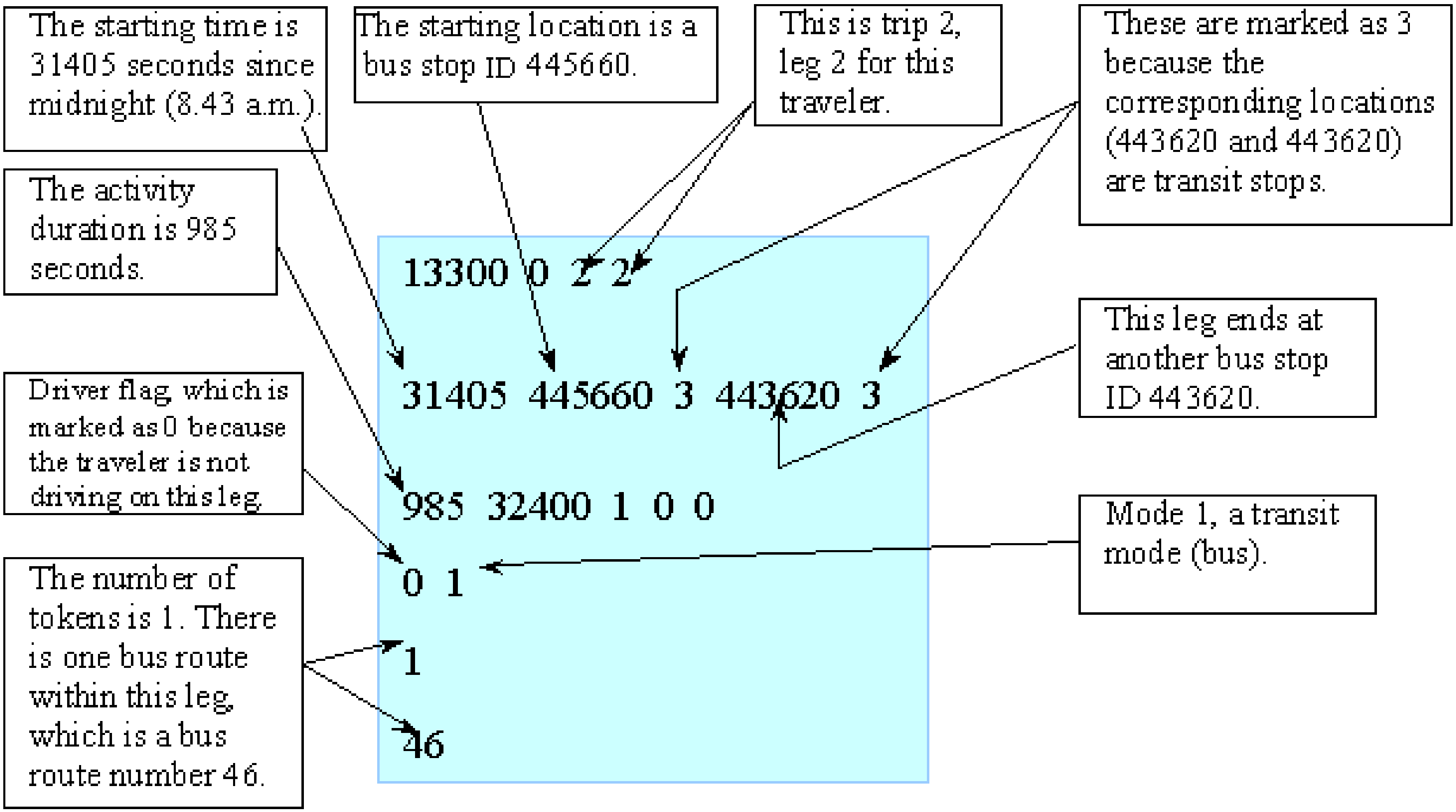
Microsimulator Input – Driving Trip Leg in the Plan File



Microsimulator Input – Walking Trip Leg in the Plan File



Microsimulator Input – Transit Trip Leg



Vehicle File and Vehicle Type File

Vehicle File

VEHICLE	HHOLD	LOCATION	TYPE	SUBTYPE
1	1	30986	1	0
2	2	82983	1	0
3	3	28612	1	0
4	4	65933	1	0
5	5	63879	1	0

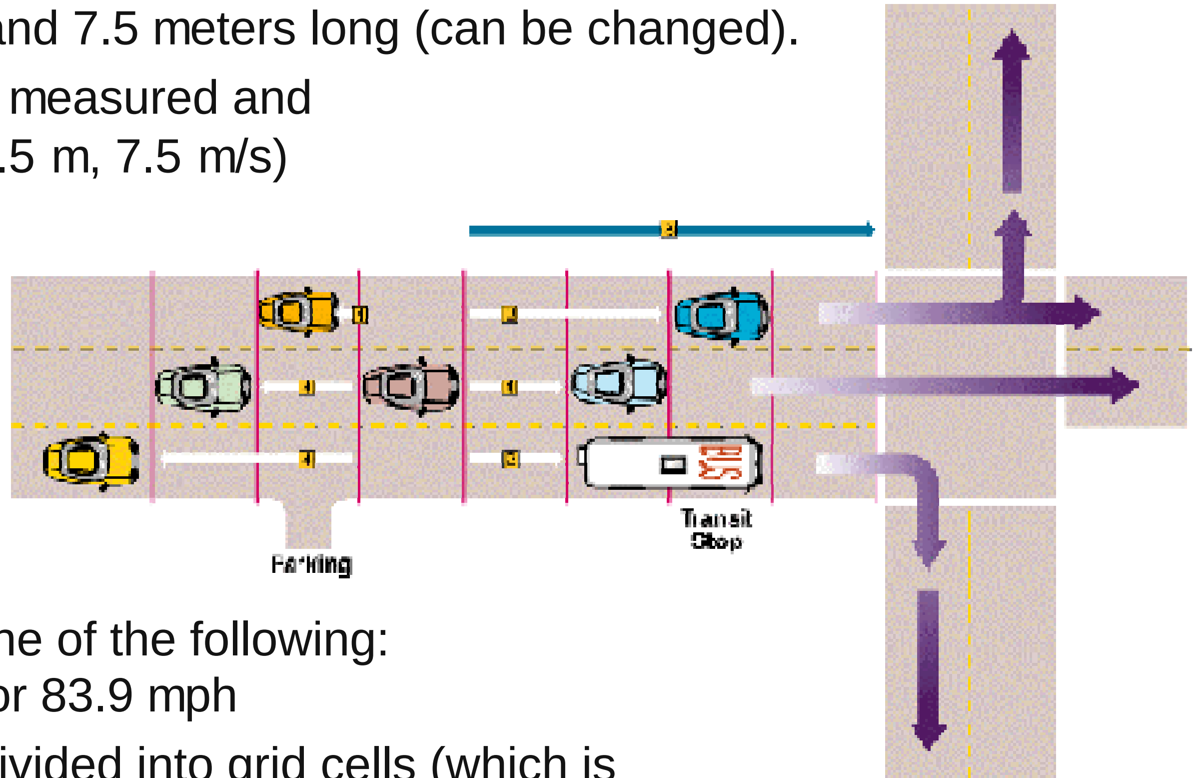
Vehicle Type File

VEHTYPE	VSUBTYPE	MAXVEL	MAXACCEL	LENGTH	CAPACITY
1	0	37.5	7.5	7.5	5
1	1	37.5	7.5	7.5	2
5	0	37.5	1.1	7.5	52
8	0	37.5	5	7.5	255

- Additional Vehicle Type Columns:
  - **MAX_DECEL**: Maximum deceleration
  - **USE**: Defining which links or lanes the vehicle is allowed on
  - **LOADING** and **UNLOADING** times per passenger
  - **MIN_DWELL** and **MAX_DWELL** times for transit vehicles
  - **METHOD**: undocumented at this time

Internal Representation of the Network

- Each roadway section is divided into grid cells, each is one lane wide and 7.5 meters long (can be changed).
- Position and speed are measured and incremented by cells (7.5 m, 7.5 m/s)



- Vehicle speed will be one of the following: 16.8, 33.6, 50.3, 67.1, or 83.9 mph
- Intersections are also divided into grid cells (which is different from Version 3 of TRANSIMS)
- Vehicles accelerate based on the vehicle type's acceleration rate or a minimum of one cell per time step.

Simulation of Vehicles and Cellular Automata

Realistic traffic behavior is simulated in the Microsimulator by making decisions about:

- Lane changes
- Passing slow vehicles
- Interactions with other vehicles (change speed, stop, etc)
- Interactions with traffic controls (signs, signals, etc.)

Decisions are based on the state of every other vehicle in its local vicinity at the same time, as well as on traffic signals and road conditions. The decisions are based on a set of simple rules.

Purpose and Functions

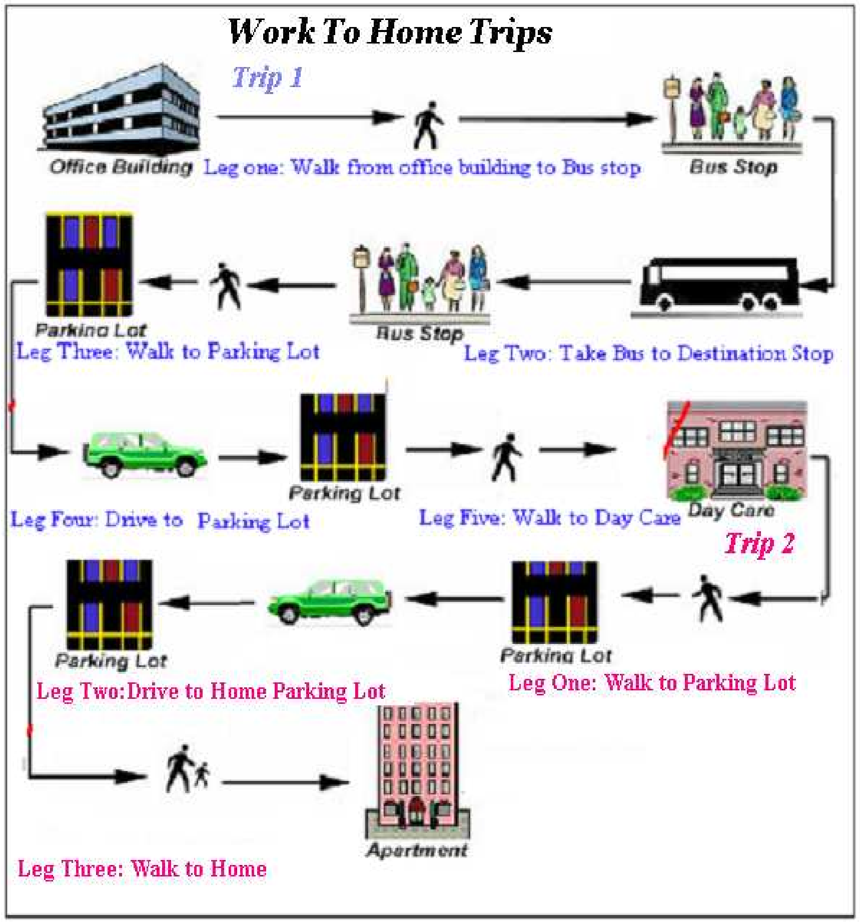
- Microsimulation results form the basis for determining network performance and environmental calculations, and for conducting the iterative feedback process with the router

- The traffic microsimulator DOES explicitly simulate:
  - Multiple travelers per vehicle
  - Multiple trips per traveler
  - Vehicles with different operating characteristics
  - Intermodal travel plans

- The microsimulator DOES NOT simulate:
  - Walking legs
  - Interaction between cars and pedestrians
  - Rail road crossings

Trip Legs – Walking and Buses

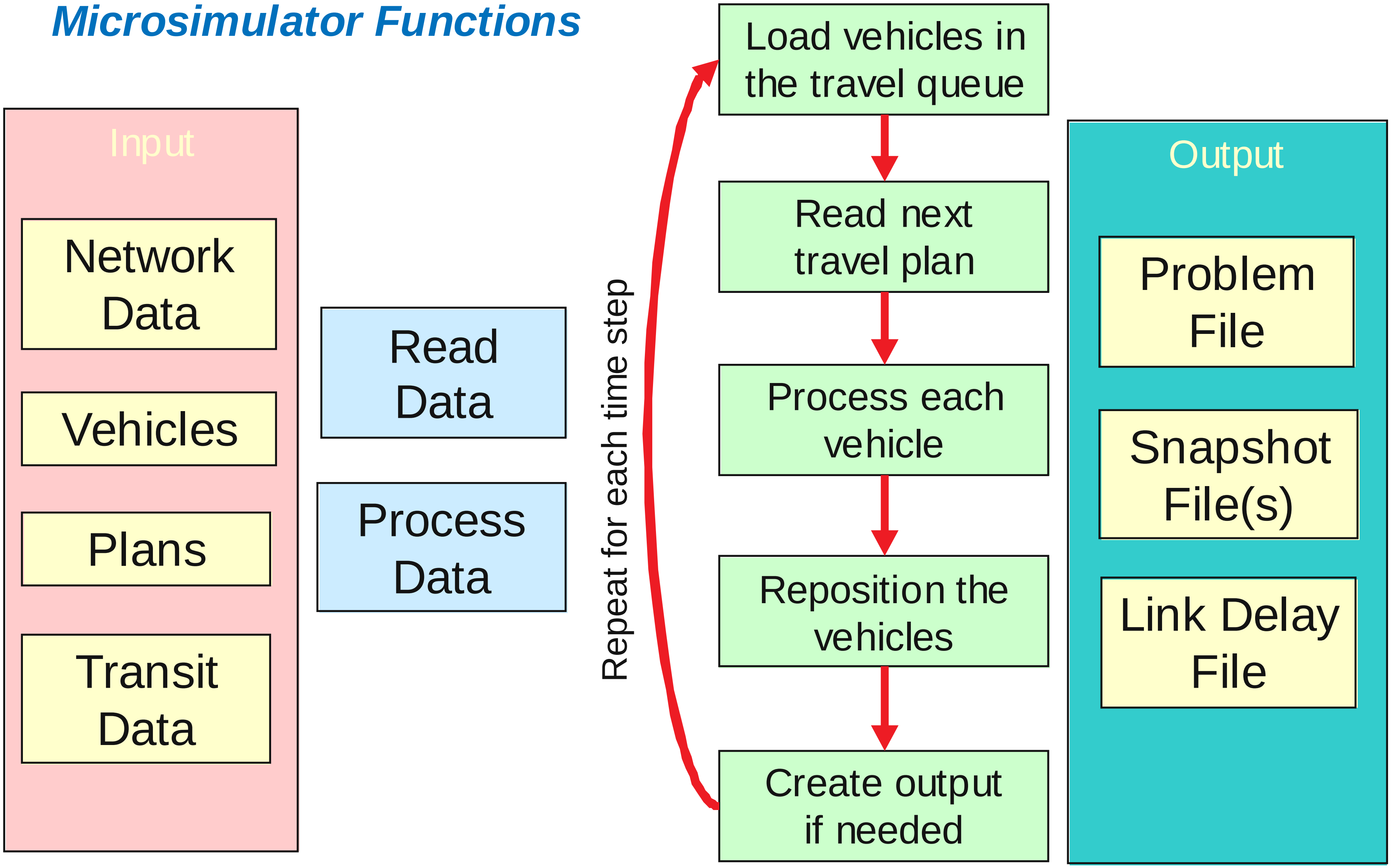
- Walk legs are not explicitly simulated
  - They begin at the time specified at the starting point of the leg
  - The traveler arrives at the destination point at a time computed by adding the delay time (process link time) to the starting time of the walk leg
- Bus legs are simulated including
  - Loading and unloading
  - Capacity and resource constraints
  - Traveler boards at bus stop, rides it until their desired stop, and exits
  - Itinerary of the bus determined by the bus schedule
  - Travelers plans require information on acceptable bus routes



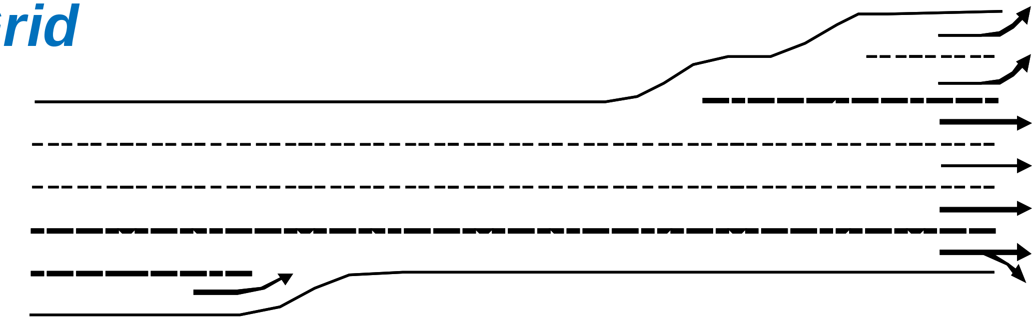
Trip Legs – Vehicles

- Vehicular traffic is fully simulated using cellular automata
  - This includes the movement of buses and other transit vehicles sharing the roads
- Vehicles start and end at parking locations (except transit vehicles)
  - External traffic is brought in through special boundary parking lots placed on the edges of the road network
- Each traveler is associated with a specific vehicle
- The vehicle must be present or the traveler must wait for it to arrive
- Plans dictate the exact movement along the network by specifying turns to make and which route to follow
  - The microsimulator follows the plans and determines the time it takes based on interaction with other vehicles, traffic signals, and roadway limitations
- Plans also specify if there is a pick-up to be made
  - Ride sharing is optimized for members of a single household. “Magic moves” allow for the sharing of rides between unrelated travelers.

Microsimulator Functions



The Microsimulation Cell Grid



- The network is physically converted to a cell grid for simulation
- Each direction of each link that permits vehicle movements (the use code includes car, truck, bus, and/or rail) and includes at least one travel lane is converted to a cell grid
- Two cell grids are created for each direction, one each for the current and the next time step
- The cells contain the vehicle identifiers of the vehicles currently occupying the cell
- Travel cells start at a "Setback Distance" and are subject to lane restrictions and traffic controls

Cell Grid

Lane \ Cell	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1																	
2																	
3																	
4																	
5																	
6																	
7																	

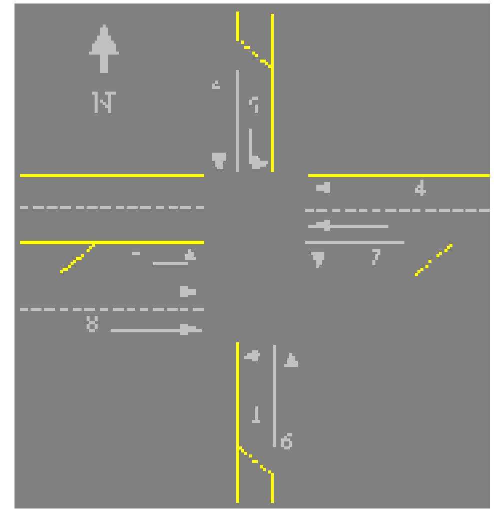
Cell Grid

Lane \ Cell	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1						
3																	
4																	
5																	
6																	
7						-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

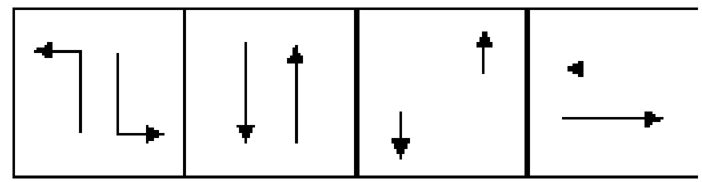
Link Connections, Thru Lanes

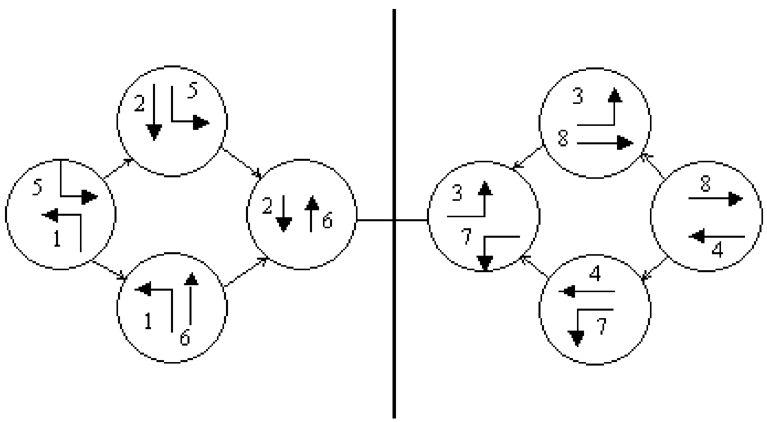
- The lane connectivity file defines a connection between a specific lane on an approach link to a specific lane on a departure link.
- The Microsimulator summarizes this data into a single link connection record with the directional approach and departure links as the record keys.
- The input and output lane numbers are derived directly from the lane connectivity file.
- The input and output thru lanes are a subset of these lanes that would not require lane merging or diverging to complete the movement.
  - In other words, if the number of input and output lanes is different, the thru lanes identify the lane numbers that are directly aligned with each other.
  - Movements between links that use these lanes are given priority over movements that use the other available lane connections.

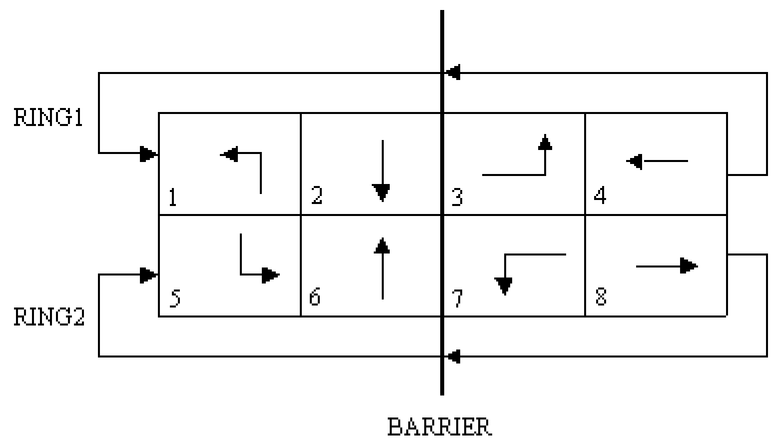
Traffic Controls



- Converting the traffic signal and sign files into a coordinated sequence of events linked to specific movements within the network is a complex process
  - Verify signal timing plans
  - Identify the phases (the sets of concurrent movements)
  - Count the number of rings
  - Check ring boundaries …







Transit Vehicles and Transit Plans

- Previous to Version 4 of TRANSISM, transit vehicles were treated like any other vehicle
  - *A driver travel plan was created for each run of each route. These plans list the origin and destination parking lot, the trip start and end times, and the travel itinerary (node list) for the trip. Transit vehicles were added to the vehicle file and initially located at the origin parking lot.*
- Version 4 has been designed to simplify the transit network development, at the cost of extra complexity within the Microsimulator itself
  - A driver path lists the sequence of links that a specific transit route uses, and the same set of links is used for each run
  - Transit vehicles are synthetically generated for each run on such a route and come into existence at the first stop of the route and cease to exist at the last stop
  - Synthetic transit vehicles are assigned vehicle identifiers at a number higher than all real vehicles and the number is listed in the standard output of the Microsimulator

Transit Vehicles and Transit Plans

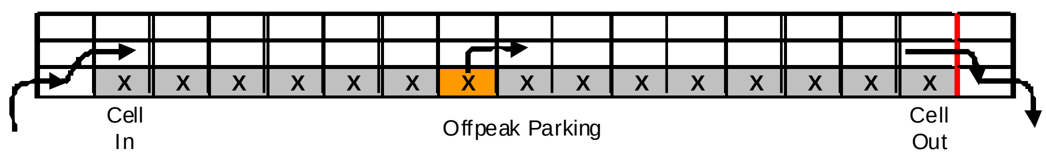
- The synthetic vehicle's identifier also contains an encoding of the route identifier and the run number.
  - The algorithm is extensively documented and goes beyond the scope of this course
- Each transit driver record is the converted into a regular travel plan.
  - They start and end at transit stops instead of parking locations
  - They have lane control points in the middle of links to force them into the right-most lane when serving transit stops for for loading and unloading
- There is a single travel plane per route that serves as a template. Real plans are derived on the fly by allocating them in an array derived from the template

Time Step Processing

- Loop through each time step (defaults to 1 step per second)
  - When the time step equals a whole second
    - *Process output increments for the second*
    - *Update the network use restrictions for the second*
    - *Update the signal timing plans for the second*
    - *Initiate transit runs scheduled to start during the second*
  - Process vehicles in the priority queue
  - Process lane change requests
  - Process the vehicle loading queue
  - When the time step equals a whole second
    - *Read travel legs from the plan file that start at the second*
    - *Convert the travel leg to a link-lane travel plan*
    - *Load the vehicle to the network or the travel queue*
  - Calculate movements for each traveler and transit vehicle
  - Move each vehicle to its new location and travel speed

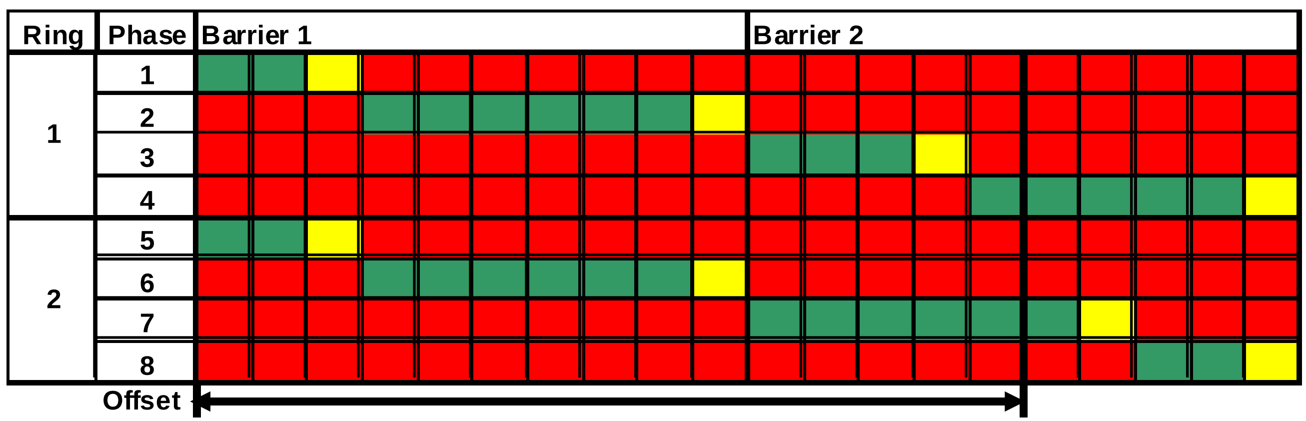
Updating Use Restrictions

- If a time-of-day restriction is found, each lane is reset to the default link restriction and then each time-of-day restriction modifies the default values based on its time, lane, and cells ranges.
  - In this process, lane-use records provided in the lane-use file cannot make a lane less restrictive than the default link use restriction.
- Since restrictions do not apply to setback cells, the lane connectivity is not directly impacted by closing or restricting travel cells.
  - Vehicles may need to immediately change lanes to move out of the setback cells to continue their trip, but the travel paths are not "broken" by adding or changing lane-use restrictions by time-of-day.
- If all of the lanes are blocked or a vehicle is on the link when the restriction takes effect, the vehicle will attempt to move out of the restricted lane or wait until the restriction is removed.



Updating Traffic Signals

- Traffic signals are initialized for
  - The first time step
  - Or when the phasing/timing plan changes by time of day
- Update cycles are based on the timing events set by the initialization
- Demand-actuated signals require some extra logic to check for vehicles on the detectors
- Barriers are checked to ensure that incompatible phases don’t occur at the same time
- Signal timing events can be logged in an event file for post-processing
- Changes also trigger cycle failure processing



Start of Transit Runs

- For each time step, a check is made whether a transit run begins on any specific route
- If yes, a specific travel plan is derived from the template and the transit vehicle is place on the network at a bus stop if possible
  - Otherwise, it is added to the queue so that it will be considered for entering the network in the next time step
- Since this process is initiated at the schedules departure time, no dwelling time or passenger loading time is considered at this first stop
  - The program assumes the transit vehicle waits at an off-network location while it loads passengers and is ready to enter the network at its scheduled departure time

Queues, Priorities, and the Processing of Movements

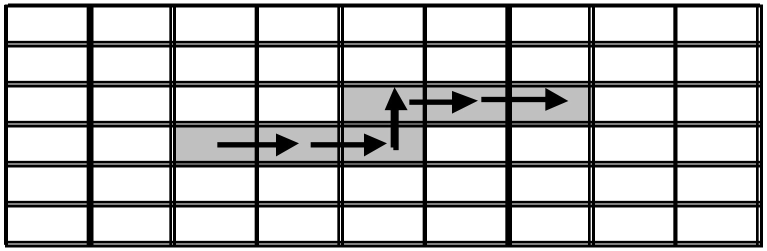
- After all of the preliminary network adjustments are made, the program loops through various vehicle queues
  - The primary queue is the network traffic queue that includes all vehicles currently on the network.
  - This queue is processed in the order in which vehicles enter the network.
- Some vehicles need to be processed at a higher priority so they have a chance to make their movements
  - The processing order is changed to give those vehicles a higher priority
  - This algorithm seals with vehicles that
    - *Have not move for a long time*
    - *That have to make lane changes to continue their trips*
    - *That have to enter the network from a parking lot*

The Priority Queue

- The first set of vehicles that are given an opportunity to move are those vehicles included in the priority queue.
  - Initially, the priority queue is empty.
  - As transit vehicles are loaded they are added to the priority queue.
  - All other vehicles move in to and out of the priority queue based on their urgency to move.
- The value of MINIMUM_WAITING_TIME is used to determine which vehicles are move to the priority queue
- A priority vehicle will be processed as a lane change or as a forward movement depending on its needs.
- The only advantage is that the vehicles that have been in the priority queue longest are given an opportunity to attempt their maneuver before any other vehicles move.
- If they fail to complete their movement, they remain in the queue and are considered a second time as part of the network traffic queue.

Lane Change Requests

- The Microsimulator maintains five lane changing queues.
  - Each queue represents a different level of lane changing urgency.
  - The vehicles in the queue with the highest urgency are given the first opportunity to attempt a lane change.
- If a lane change is feasible, the vehicle is immediately moved over into that lane in the current cell grid, and the vehicle is removed from its original cell in the current cell grid.
- If the vehicle is a multi-cell vehicle, the second cell of the vehicle will be moved up to the original cell in the original lane ("snaking behind the lead cell").



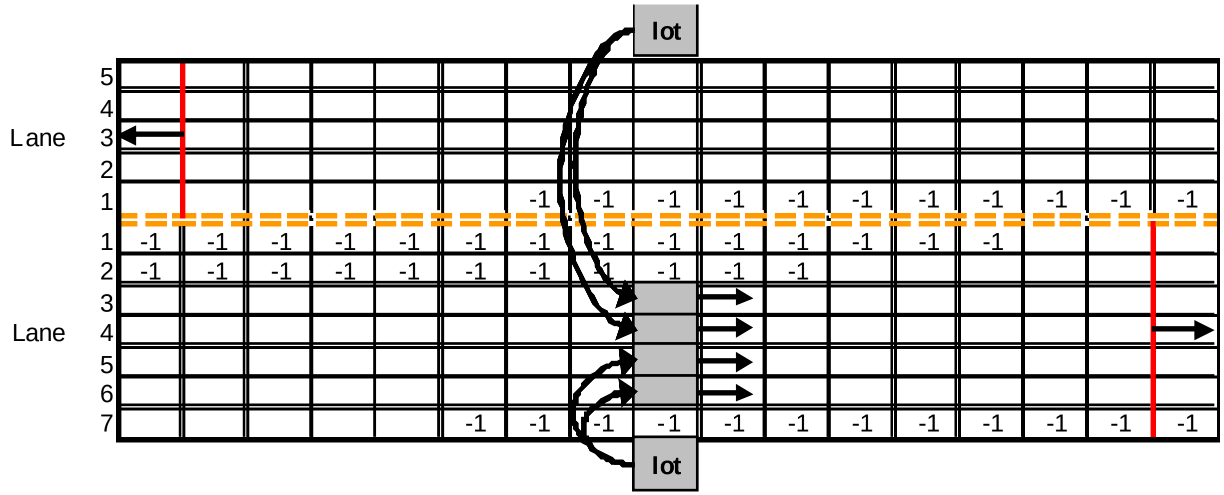
- A lane change required by network or lane-use restrictions is always added to the priority one lane changing queue.
- The priority level for plan following lane changes is a function of the number of required lane changes and the distance to the end of the link.

Actual Lane Changes

- If neighboring cells are permissible, a lane change can be made
  - The code checks for use restrictions, pocket lanes, and the location of other vehicles in these cells
  - If another vehicle is in the way, the code checks whether a “cooperative lane change” can be made
  - Lane change requests have created “reservations” for target cells that help with deciding on how to cooperatively change lanes
  - Parameters control adherence to a maximum swapping speed and a maximum speed difference
- If the target cell is empty, the algorithm checks for the distance of approaching vehicles from behind to make a decision on the feasibility of a lane change
- Extensive documentation of the algorithms is available

The Loading Queue

- Each time a travel plan involving a vehicle starts, the program attempts to immediately load that vehicle to the network.
- The vehicle is inserted into one of the lanes next to the parking lot and the travel plan is added to the network traffic queue.
- If none of the permissible loading lanes are available or the vehicle is currently in use on another trip, the travel plan is added to the travel queue and reconsidered at a later point in the process.
- For standard parking lots or transit stops or stations, the two lanes closest to the lot or stop are used. This includes pocket lanes that are available at the cell offset of the parking lot or transit stop.

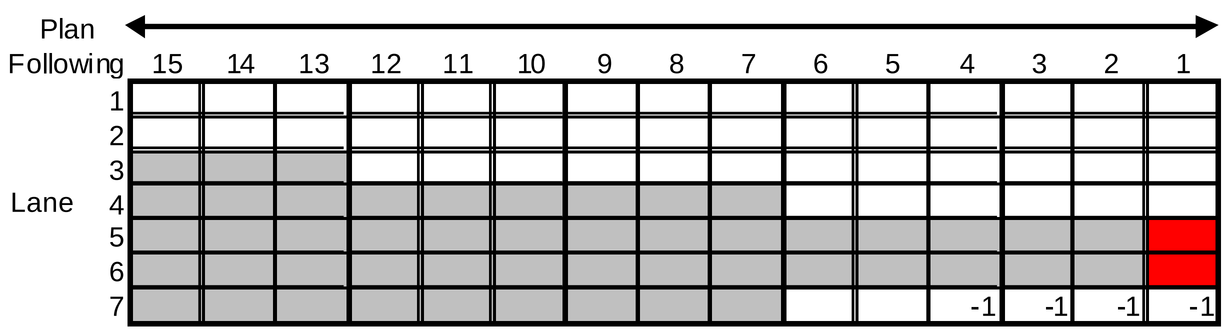


Loading Vehicles onto the Network

- The concept of loading vehicles on the closest two lanes does not consider any lane-use restrictions that make the lane unavailable at the time of the trip.
  - This is one of the primary reasons why two lanes are included in the loading process. If one lane is unavailable due to a lane-use restriction, the other lane may still be available to load the vehicle. If both lanes are unavailable, the vehicle will wait in the parking lot until the restriction is removed.
- The lane rules apply to vehicles exiting the network as well.
  - The vehicle will make lane changes to the appropriate side of the street before being pulled off into the parking lot.
  - The modeler can override this behavior by setting the ENFORCE_PARKING_LANES key to false. In this case the vehicle will attempt to move into an exit lane, but if it fails to do so before reaching the parking lot, it can be removed from any lane.
- Exceptions: Boundary parking lots and external transit stops
  - Vehicles are loaded onto all lanes and at a higher speed
- Many more details are well-documented

The Plan Following Concept

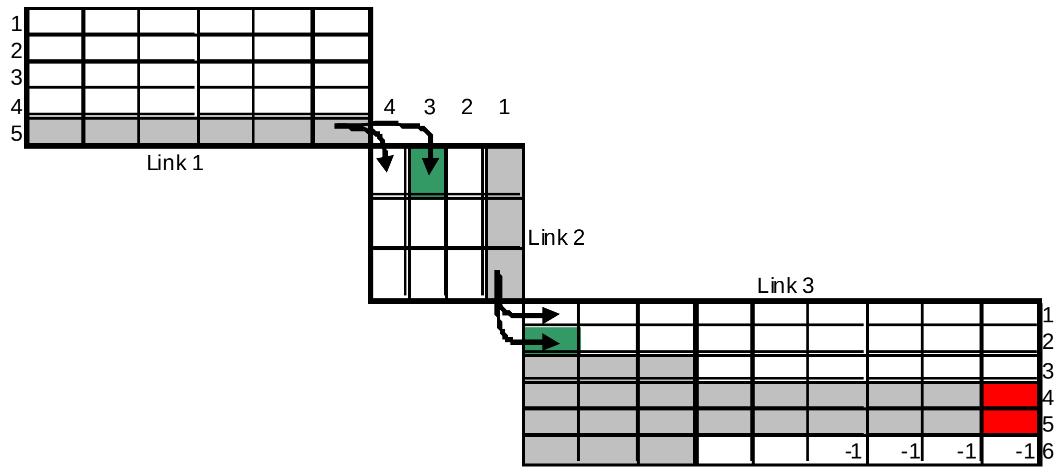
- The best lane logic starts at the end of the trip and works back to the beginning of the trip, tracking the distance and lane restrictions as it proceeds. The initial lane restrictions are defined by the destination parking lot type.
- Given these restrictions and the distance to the prior intersection, the plan following algorithm may limit the best lanes for the movement at the prior intersection to a subset of the permissible lanes.
- The plan following algorithm divides the plan following distance into five equal segments.
  - No lane changes are desirable in the last two segments.
  - The third and fourth segments are limited to one lane change
  - The first segment is limited to two lane changes.



Complexity of Plan Following Cases

- The plan following concepts can become complex when turning movements are involved. In the example below, the travel path includes three links and two turns in a distance equal to the plan following parameter. The vehicle needs to exit from lanes four or five on link 3. It enters link 3 by way of a left turn from link 2. The lane connectivity from link 2 to link 3 is from lane 1 on link 2 to lanes 1 or 2 on link 3. The plan following algorithm would flag this as a problem and select the closest turning movement as the best lane for entering link 3. This would favor the movement from lane 1 to lane 2 over the movement from lane 1 to lane 1.

(This particular example and the materials for many of the previous viewgraphs has been copied from the excellent documentation that AECOM is developing for TRANSIMS)



Discretionary Lane Changes

- Vehicles are given the opportunity to make discretionary lane changes
  - Discretionary lane changes are lane changes that are not required by the travel plan, but may help the vehicle avoid congestion or slower vehicles ahead of them in the current lane (i.e., pass a slower vehicle).
  - Like any other lane change, a lane must be available to change into before a lane change is even considered.
  - These lane changes are considered on alternating sides of the current lane.
    - *On odd time steps the vehicle considers the current lane plus one.*
    - *On even time steps the vehicle considers the current lane minus one.*

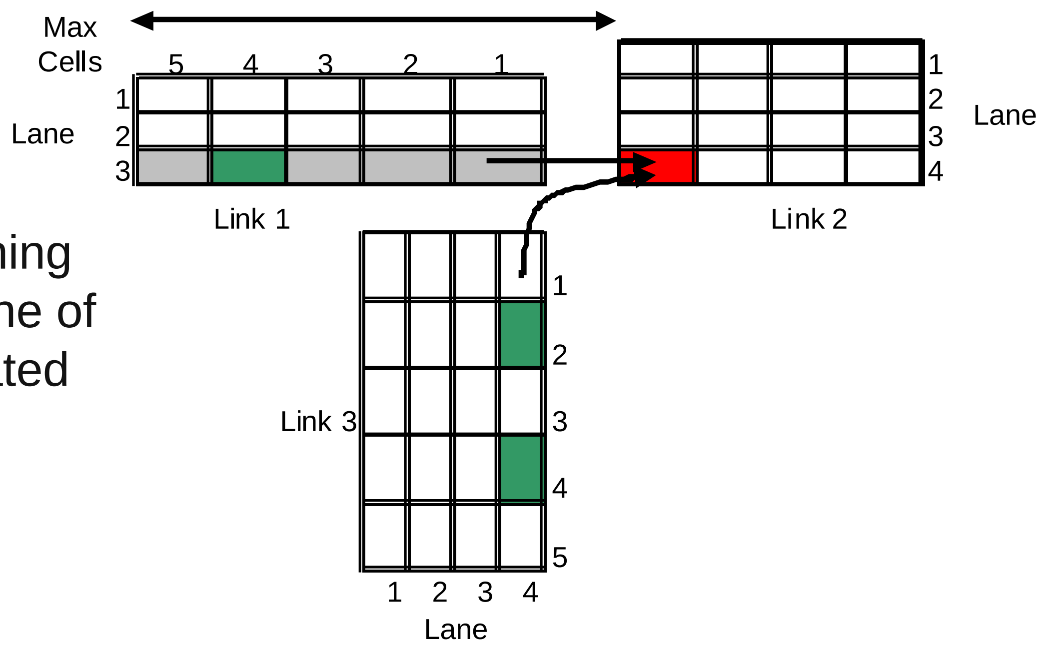
Look Ahead →

Lane	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	-1	-1	-1	-1	-1	-1	-1	-1	-1							
2																
3																
4	↱	4	5	4	5	5	5	2	3	4	5	5	5	0	0	= 55
5	3	4	3	4	5	2	3	4	5	1	2	0	0	0	0	= 36
6																
7						-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	

Max Speed = 5 cells / second

Check-Behind Algorithm

- The check-behind routine is used in lane changing calculations and forward movements to determine if an adequate gap exists for a vehicle to merge into a given cell.
- The routine starts with a vehicle in a specific location, traveling at a specific speed, and the number of time steps ahead that the vehicle is expected to arrive at this location.
- The routine scans the cells upstream from the current cell to locate the first vehicle or pocket lane it encounters.
- If this search reaches the beginning of the link, it continues on the lane of the upstream link that is designated as the priority thru movement.



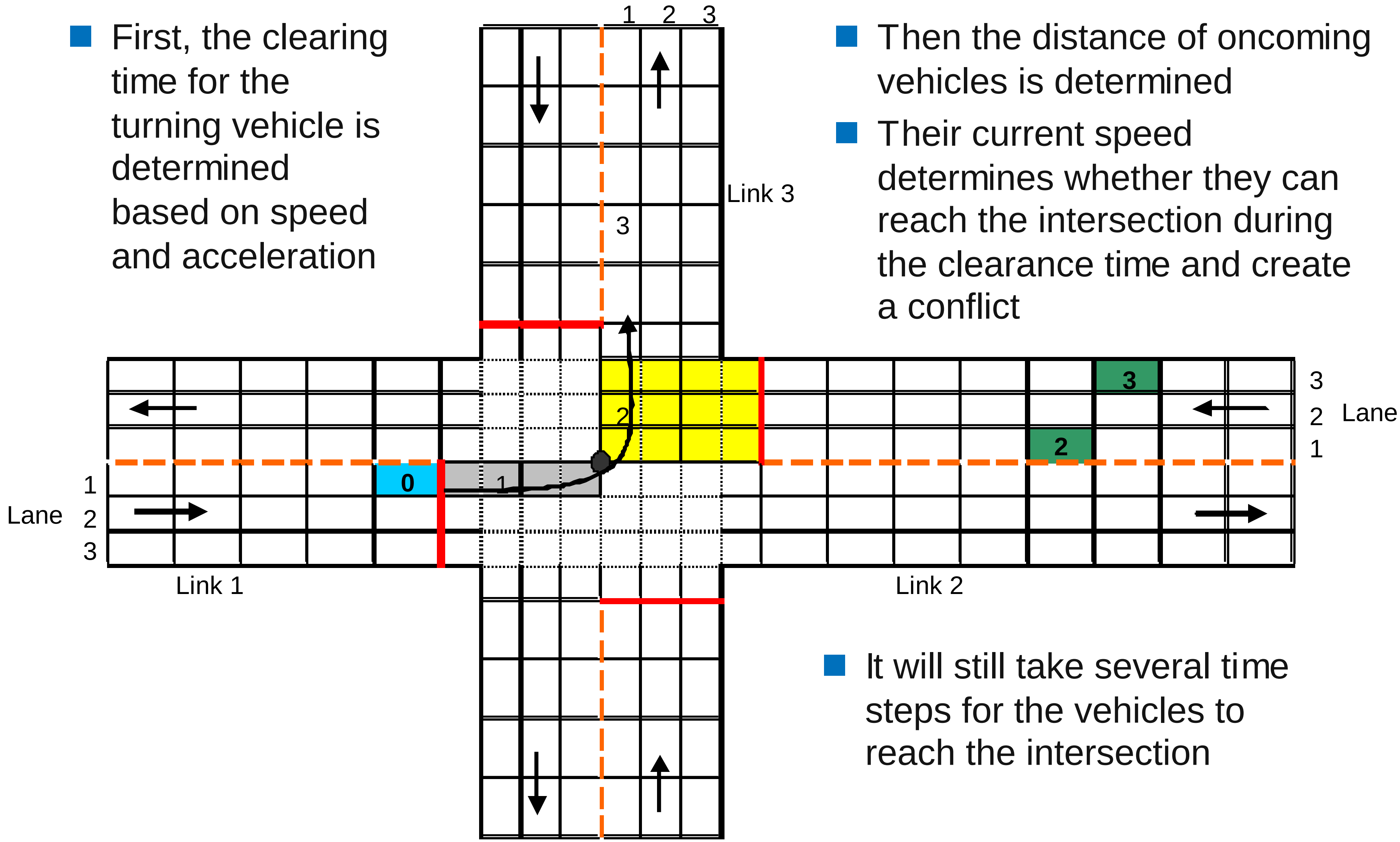
Check-Ahead Algorithm

- All forward movements for a vehicle are initially set by the check-ahead routine. This routine can be called in test-mode or save-mode.
- The output of the check-ahead routine is the number of cells the vehicle can move during the time step (i.e., maximum speed) and flags for lane changing and simulation problems.
- The program then checks if the vehicle is at the link-offset of the destination parking lot.
  - If it is and the ENFORCE_PARKING_LANES key is true, the current lane is checked against the exit lane range.
  - If the vehicle is not in the appropriate lane, the speed is set to zero and a lane change request is issued.
- If the vehicle is located in the last travel cell on the link and the link has traffic controls, the control for the appropriate link connection is retrieved and its status checked.

Intersections and Traffic Controls – Oncoming Traffic

- First, the clearing time for the turning vehicle is determined based on speed and acceleration

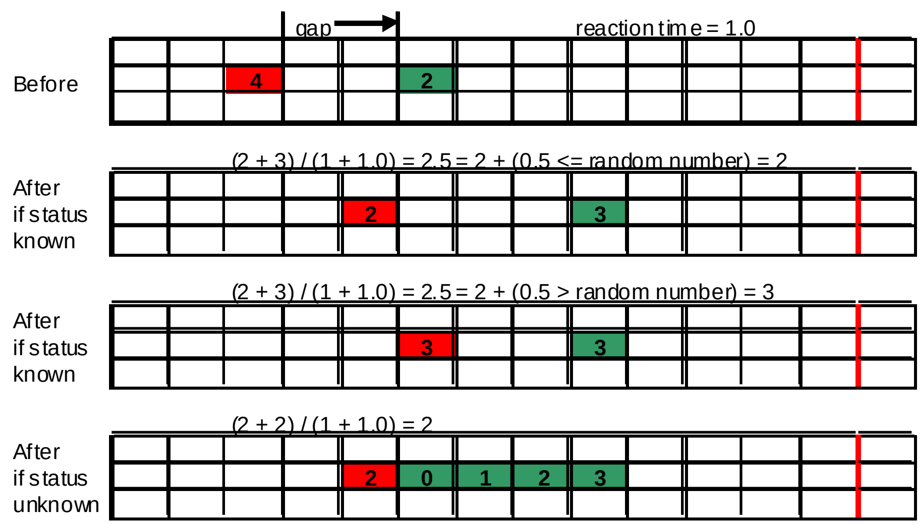
- Then the distance of oncoming vehicles is determined
- Their current speed determines whether they can reach the intersection during the clearance time and create a conflict



- It will still take several time steps for the vehicles to reach the intersection

Reaction Time Concepts

- The speed of the vehicle ahead plus the gap between the current vehicle and the vehicle ahead is used in the reaction time calculation.
- Reaction times can be specified by facility type
  - The default reaction time for all facility types is 1.0 seconds
- If the vehicle ahead had previously been processed during this time step, the result of its movement is considered by the vehicle behind. If it has not been processed, its current speed is considered. The speed of the vehicle ahead plus the gap between the current vehicle and the vehicle ahead is used in the reaction time calculation.
- This calculation allows for vehicles to follow each other closely. In Version 3, vehicles could only move into cells that were empty in the previous time step



Basic Parameters and their Default values

CELL_SIZE	7.5
TIME_STEPS_PER_SECOND	1
TIME_OF_DAY_FORMAT	SECONDS
SIMULATION_START_TIME	0:00
SIMULATION_END_TIME	26:00
PLAN_FOLLOWING_DISTANCE	525
LOOK_AHEAD_TIME_FACTOR	1.0
LOOK_AHEAD_LANE_FACTOR	4.0
LOOK_AHEAD_DISTANCE	260
DRIVER_REACTION_TIME	0.7
RANDOM_NUMBER_SEED	1122687672
MINIMUM_WAITING_TIME	60
MAXIMUM_WAITING_TIME	120
MAX_DEPARTURE_TIME_VARIANCE	60
MAX_ARRIVAL_TIME_VARIANCE	60

Microsimulator Output Files

- **Summary Files**
  - Aggregates volumes and performance statistics (e.g., travel time) for each link direction and turning movement by time increment (e.g., 15 minutes).
  - Turning movement volumes and delays are optional.
- **Snapshot**
  - Lists the link direction, offset, lane, and speed of each vehicle at specified time points (e.g., every 5 minutes)
- **Occupancy**
  - Lists the link direction, offset, lane, and cumulative occupancy of each cell by time increment (e.g., 15 minutes). May list the cells occupied at the maximum load point during the time increment or the total occupancy of each cell during the time increment
- **Ridership**
  - Summarizes the boardings and alightings at each stop on each route based on the scheduled and actual departure time for each run.

Microsimulator Output Files

- **Event**
  - Lists the scheduled and actual time and link direction and offset for each traveler and trip event (i.e., start time and end time).
- **System Event**
  - Lists the time, node, and phasing information for each phase or timing plan change event at a traffic signal
- **Problem Link**
  - Aggregates the number of problems by problem type (e.g., wait time) for each link direction by time increment (e.g., 15 minutes).
- **Traveler**
  - Lists the link direction, offset, lane, and speed for each selected traveler by time step (e.g., second)
- **Turn**
  - Aggregates the number of turning movements at a node by input and output link and time increment (e.g., 15 minute)

Microsimulator Output Files

- **Speed Bin**
  - Aggregates the number of vehicles of a specified vehicle type by speed bin traveling on link segments at specified time increments
- **Problem**
  - Lists time, link direction, offset, and lane where trips experience problems in the Microsimulator. The output includes data about the trip (e.g., household, person, trip, mode, origin, destination, and start and end times).

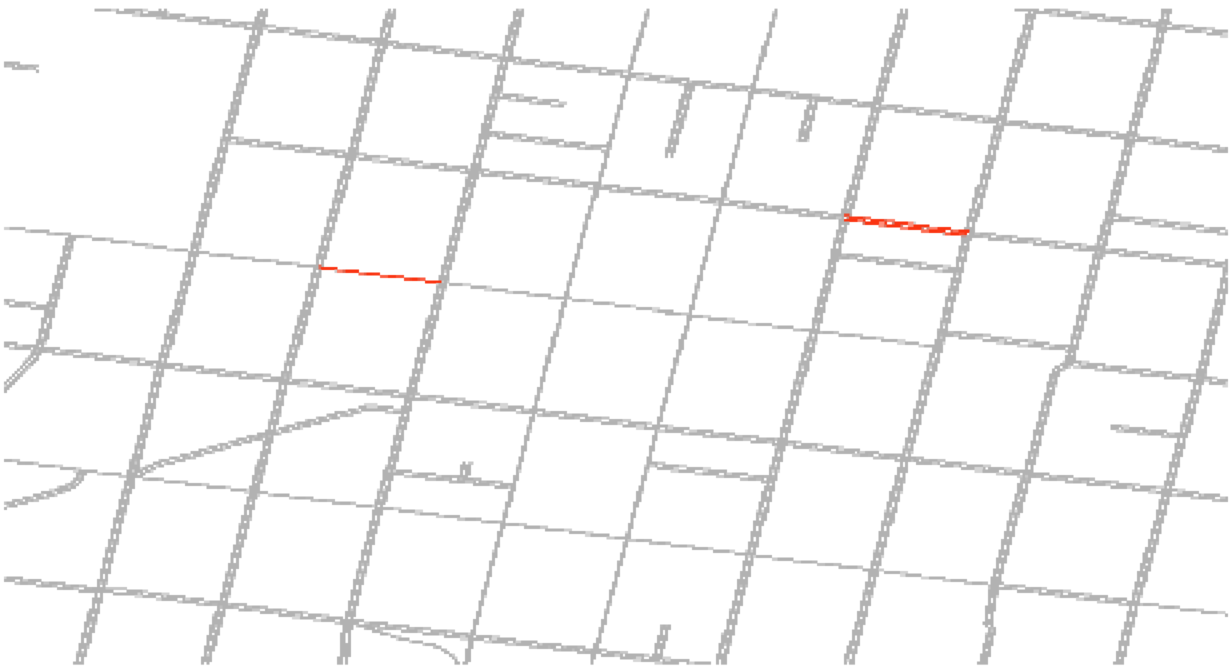
- Extensive documentation of all the Microsimulator output files can be found on the TRANSIMS Open Source Web Site at:

http://transims-opensource.org/wiki/doku.php?id=software:version4:microsimulator_output_file_formats

Problem File

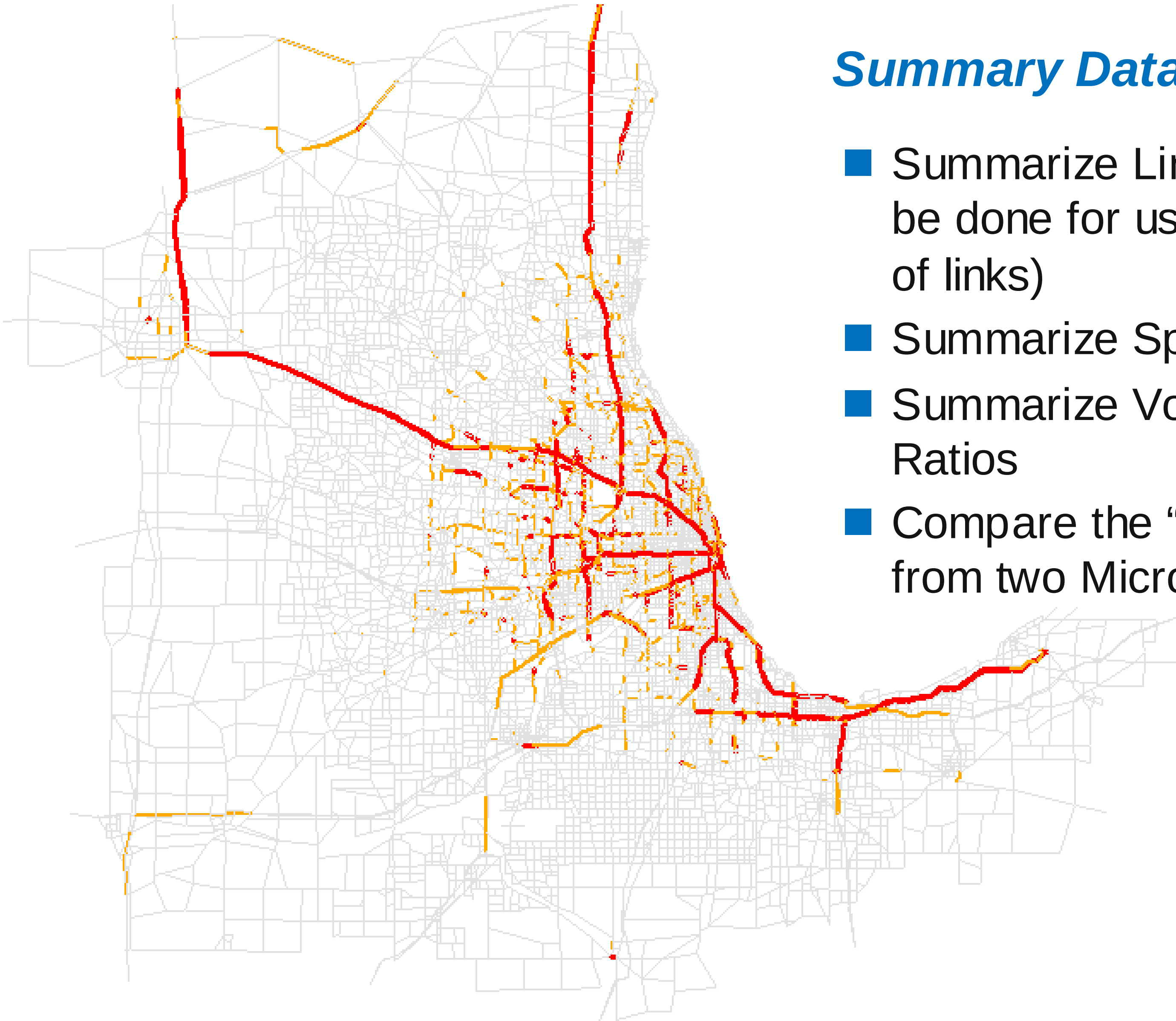
- If congestion or network errors prevent a vehicle from changing lanes as necessary within a user-specified time period, the vehicle will be removed from the network and considered “lost.”

- A vehicle that is removed from the simulation typically points to a problem (geography, unrealistic route)



Snapshot Data Example

- Location of all vehicles for every second if requested
- Can be visualized in form of GIS shape files as shown
- Animation in form of video sequences has been routinely used, and will be improved upon



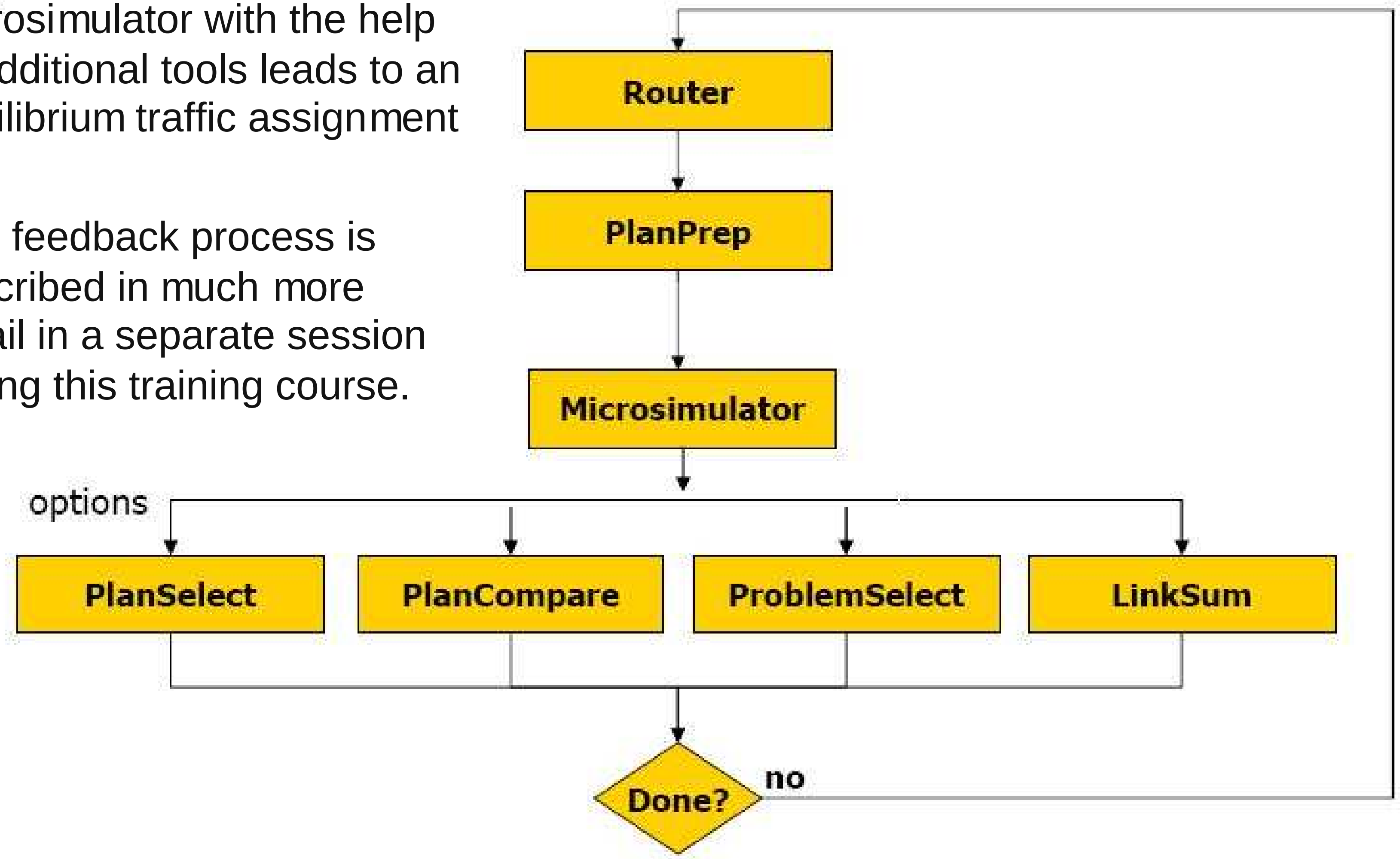
Summary Data Example

- Summarize Link Volumes (can be done for user-defined group of links)
- Summarize Speeds
- Summarize Volume-to-Capacity Ratios
- Compare the “Link Delay” Files from two Microsimulator Runs

Feedback Process

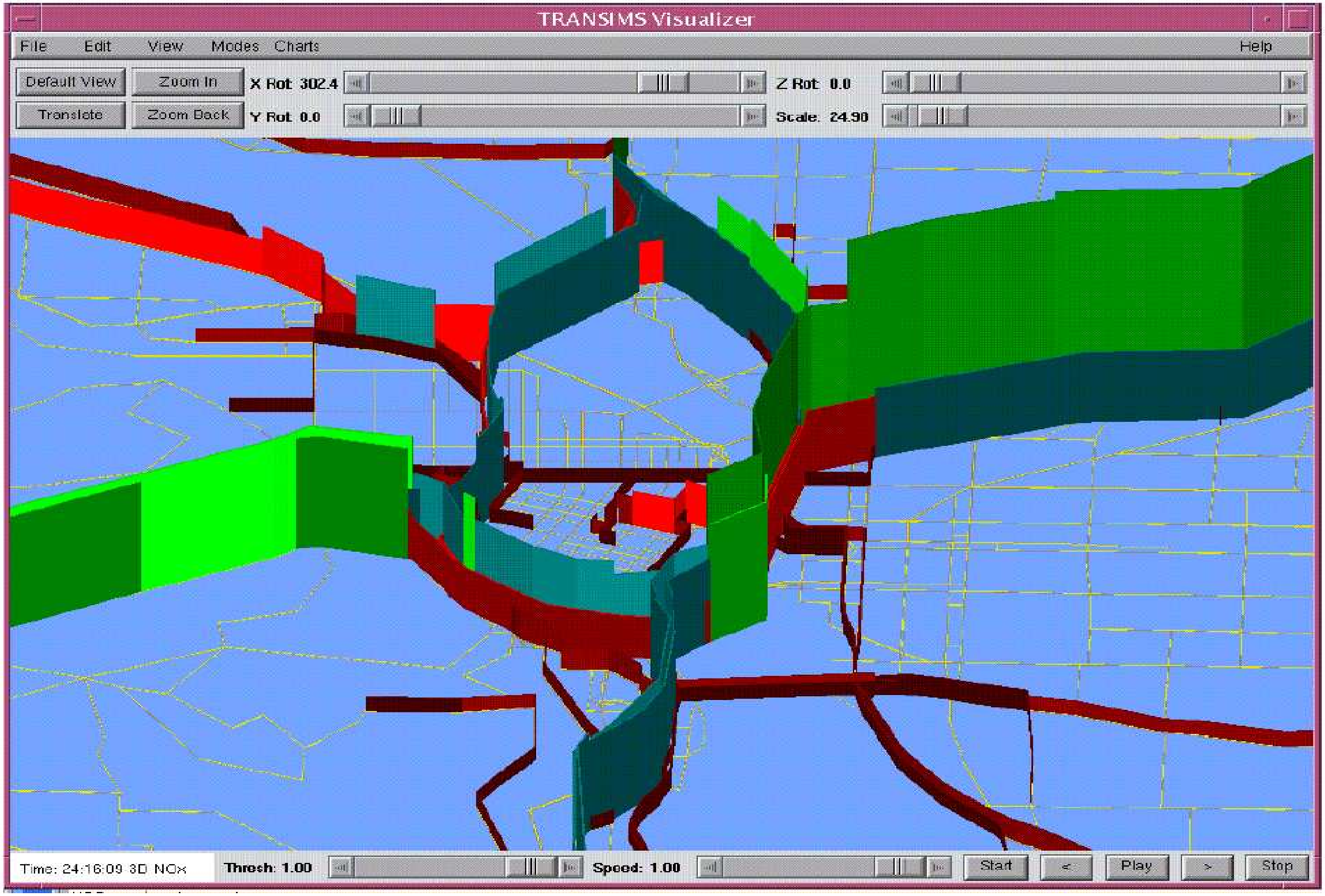
- Iterations between router and microsimulator with the help of additional tools leads to an equilibrium traffic assignment

- The feedback process is described in much more detail in a separate session during this training course.



Specific to Version 1

Visualization of Microsimulator Output



Specific to Version 1

Visualization of Microsimulator Output

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 and to a much more significant degree on the Microsimulator documentation by David Roden from AECOM