Dynamic Transit Simulation

Innovations in Travel Modeling 2008

David B. Roden
Senior Consulting Manager
AECOM Consult
3101 Wilson Boulevard, 4th Floor
Arlington, VA 22201
david.roden@aecomconsult.com
703-682-5079

The White House Area Transportation Study is evaluating mitigation strategies designed to relieve traffic congestion in downtown Washington D.C. resulting from the security closures of E Street and Pennsylvania Avenue in 2001. The study is being conducted by FHWA in consultation with the National Capital Planning Commission, the District of Columbia, the National Park Service, the U.S. Secret Service and WMATA. The technical analysis involves a detailed simulation of the traffic and transit operations in downtown Washington D.C. and northern Virginia.

The transit simulation network uses a stop-schedule approach where each transit route is defined as a series of stops and the service provided by the route is defined by a schedule. The schedule defines when a transit vehicle serves the stop throughout the course of the day. It also defines the travel time between stops.

From the transit passenger's perspective, the trip starts by walking to a stop; waiting for the next scheduled vehicle to arrive; boarding the vehicle, paying the fare and riding to the destination stop; alighting and walking to the destination or another transit stop. If the trip includes park-&-ride, the traveler will walk to their car, drive to a park-&-ride lot, park their car, walk to the transit stop, and continue as before. The car will remain at the park-&-ride lot until the traveler or another household member returns to retrieve it.

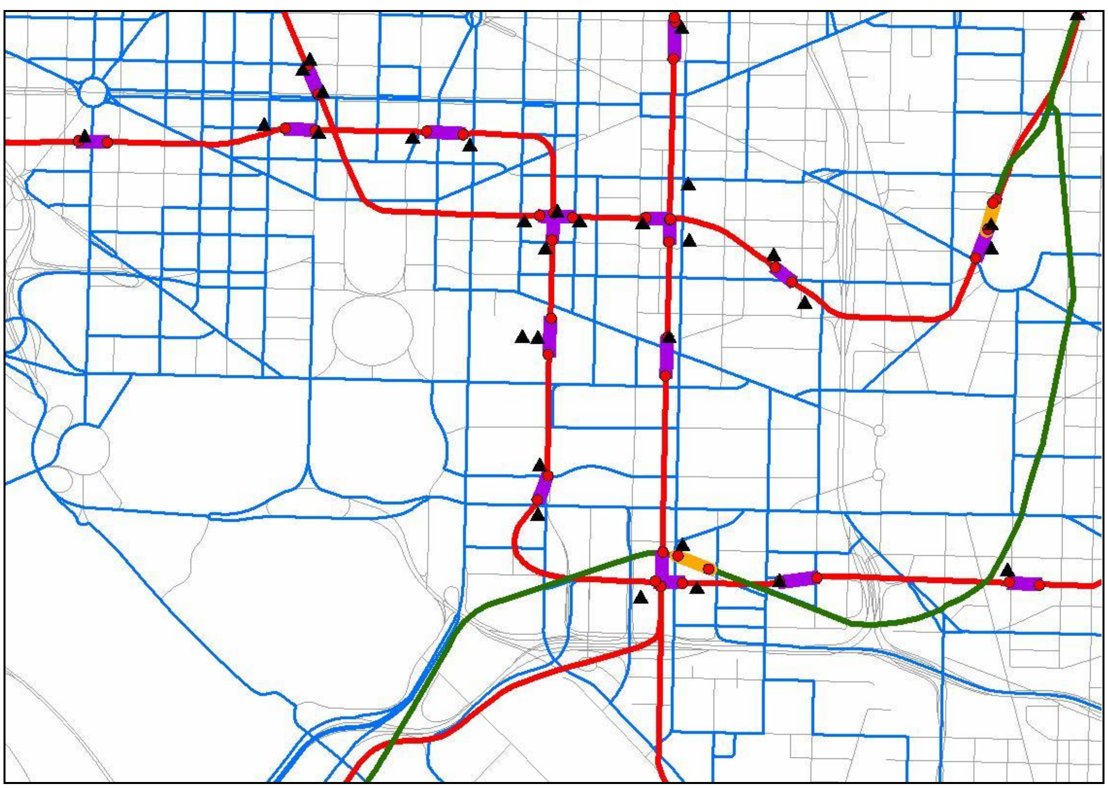
From the transit vehicle's perspective, the path between stops is defined as a series of links in the simulation network with BUS or RAIL as one of their vehicle use options. The transit driver file stores the link path and defines the vehicle type. The vehicle type file defines the size, capacity, speed, acceleration/deceleration, boarding/alighting rates, and maximum/minimum dwell times for each vehicle type. The software automatically generates a vehicle of the specified type for each run of each transit route and moves this vehicle through the network based on it interactions with all other vehicles and its schedule constraints.

The network generation process addressed most of the problems associated with physically integrating the transit routes with the detailed highway network. The next step was for transit travelers to find logical paths through the network. Access checks between the walk network and the transit network were performed. The impedance parameters were calibrated to generate logical transfers between routes and park-&-ride paths were evaluated to ensure that auto trips could access the park-&-ride lots and walk paths could be built between the park-&-ride lots and the transit network.
One routing problem related to the connection of the MetroRail system to the walk network. In order to board or alight at a transit stop, the traveler needs to be able to walk to or from the activity location attached to the stop. If this activity location is attached to a link that does not permit

walking, the stop cannot be used to transfer to other transit lines or walk to destination activity locations. Since all stops need to be offset from the intersection node, station stops and transfers at cross streets will always require a short walk to access the route. If the route or the cross street are on exclusive guideways, special coding techniques are needed to facilitate the connection. This also affects the way park-&-ride lots are coded. A park-&-ride lot must be on a link that permits both auto and walk access. If this is not the case, special access links must be added to enable the connection between the park-&-ride lot and the transit stop.

This problem was addressed in one of three ways. The first solution changed the link number on the activity location generated for the MetroRail station to a cross street link that permitted walk access. This enabled the Router to access the station by walking along the cross street to another transit stop or the trip origin or destination. If the cross street included a near-by bus stop, it was often easier to change the access links connected to the MetroRail station to use the activity location attached to the bus stop rather than relocated the MetroRail activity location. This provided a direct connection between the bus route and the MetroRail station.

The solutions outlined above were appropriate for most MetroRail stations, but oversimplified the access considerations for the subway transfer stations in downtown D.C. Most of the downtown stations include multiple access points from different directions. At transfer stations it is also possible to access the platform for one line and walk to the platform for the other line. Both of these considerations were coded into the downtown MetroRail network shown in the figure below. First, a link was coded for each platform based on its actual length and location. This link was code with rail and walk access permissions. The links between platforms remained as rail-only to prohibit passenger from walking between stations on the guideway. Next, a transit activity location was placed on the street level network at the actual location of each MetroRail access point. Process links were included to connect these activity locations to the appropriate platform location. Finally, process links were added between platforms at transfer stations to replicate the transfer path.



The Router provides several keys for controlling the relative weights of various aspects of a transit trip. These weights needed to be calibrated to create a logical balance between walking, waiting, transferring, and riding in the Washington subarea. Over 50 Router runs were made to test various combinations of transit path impedance parameters. The set of parameters that generated the most reasonable results are listed in the table below.

Router Control Key	Value
WALK_TIME_VALUE	10 units / second
FIRST_WAIT_VALUE	20 units / second
TRANSFER_WAIT_VALUE	20 units / second
VEHICLE_TIME_VALUE	10 units / second
COST_VALUE	29 units / cent (= $12.42/hr)
TRANSFER_PENALTY	300 units (= 30 seconds)
RAIL_BIAS_FACTOR	0.8 * travel time
RAIL_BIAS_CONSTANT	-100 units
MAX_WALK_DISTANCE	3000 meters (~1.86 miles)
MAX_WAIT_TIME	60 minutes
MAX_NUMBER_OF_TRANSFERS	3
MAX_NUMBER_OF_PATHS	4
MAX_PARK_RIDE_PERCENTAGE	35 percent

There are a number of interesting and somewhat unexpected findings implied by the values in the table above. The first is that the value of walking is equal to in-vehicle time. Most travel demand forecasting models weight walking at 2 to 2.5 times in-vehicle time. Combining this walk value with the maximum walk distance enabled the software to constrain walk trips to reasonable lengths, but also minimize the number of transfers involving short transit trips. (Note: the maximum walk parameter is the cumulative walk distance for all walk legs on the trip.)

Another interesting outcome is the small transfer penalty. Travel demand forecasting models frequently penalize transfers by 5 to 10 minutes. In this case a 30 second penalty combined with the transfer wait time and the maximum number of transfers kept the number of transfers to a reasonable level. This is partially due to the fact that schedules are used so the waiting time for any given transfer can be much more significant than traditional models would estimate.

The rail bias factors for MetroRail are significant, but not unexpected. The implication is that travelers perceive MetroRail as 20 percent faster than it really is. This partially reflects the volatility of the travel times for buses caused by highway congestion.

The park-&-ride percentage set to 35 percent is somewhat surprising. There were some very long park-&-ride trips from the outer suburbs, but in general the trips to park-&-ride lots appeared to be relatively short compared to the overall trip length.

Once a logical set of paths were generated, the Microsimulator was applied to the transit network to identify additional coding problems. The original transit vehicle simulation generated significant traffic delays due to buses waiting at transit stops. The bus would block the roadway for several minutes waiting for its scheduled departure time at each stop. This was caused by a disconnect between the transit schedules and the traffic signals. Since it was virtually impossible to synchronize the bus schedule to the signal phasing and traffic congestion, most routes were re-coded to only include a few critical time points. Buses were permitted to leave all other stops as soon as the passenger processing was completed. In other words, the bus will only stop to serve passengers and not to maintain a fixed schedule for every stop.

Another problem with the transit simulation was found processing passengers on the MetroRail system. When a bus or train arrives at a stop, riders get off, and the boarding queue is processed until the maximum load (capacity) is reached. The remaining travelers wait for the next bus or train on the route. Originally the stop delay per passenger was set as 2 seconds to alight and 3 seconds to board the transit vehicle. This was reasonable for buses, but created huge passenger processing delays on MetroRail.

Changes were implemented to include boarding and alighting rates by vehicle type. The passenger processing rates for MetroRail were then set to reflect multiple door loadings for four, six, or eight car trains. A maximum and minimum dwell time was also added for MetroRail. This forced the MetroRail trains to stop at each station for a minimum of 10 seconds and a maximum of 60 seconds. After 60 seconds the doors are closed and any passengers still waiting to board at the station will need to wait for the next train. This was needed to address situations where the passenger arrival rate at the station was higher than the boarding rate of the vehicles.

The simulation eventually generated logical transit operations, but also identified numerous locations where the assumed transit travel times built into the schedule significantly underestimated the system performance. For bus routes this was mostly caused by traffic congestions, but even the schedules for MetroRail were too optimistic. The fact that MetroRail operates at near capacity conditions in downtown D.C. means that the passenger processing delays are considerable. It is often the case that a train must stop on the tracks and wait for another train to load and unload passengers at the next station.

The simulated delays caused by traffic and transit operations were used to update the transit schedules. The transit travelers were re-routed and re-simulated in an iterative process until the travel times included in the transit schedules matched the travel times generated by the simulation. Initially, the shift was away from buses and onto MetroRail. This increased the congestion and delays on MetroRail which made the bus alternative more attractive.

Since the White House study is primarily concerns about mitigating traffic congestion, most of the alternatives included short tunnels to alleviate congestion. Both DDOT and NCPC have a general policy that discourages automobile use and encourages transit use in downtown, so new tunnels were not viewed favorably. These agencies suggested several alternatives that improved transit service. Ironically, it was often the tunnel alternatives that actually improved transit operations and the transit alternatives that increased traffic congestion. Finding the optimal balance between roadway and transit facilities that improved total system performance was truly challenging.