Veitch Lister Consulting Pty Ltd

06-12-2012

Zenith Technical Note

Static Traffic Assignment – Methodology

Travel Demand Forecasting & Transport Infrastructure Planning

Brisbane Ground Floor, 80 Jephson Street Toowong Qld 4066 **Tel: +61 7 3870 4888** **Fax:** +61 7 3870 4446
Melbourne Level 7, 423 Bourke Street Melbourne Vic 3000 **Tel:** +61 3 9602 5200 **Fax:** +61 3 9602 5166

www.veitchlister.com.au

Zenith Technical Note

Static Traffic Assignment – Methodology

Date	Revision	Prepared By	Checked By	Approved By	Description
24-10-2012	A	MJin / TV	AMA	TV	Draft Report
06-12-2012	B	AA	AA	AA	Updated report cover

Contents

1 Introduction

This Technical Note is one of a series of papers that collectively describe the Zenith Travel Model. Zenith is a classical four step model, implemented in the OmniTRANS software package for a range of Australian cities and regions.

This Technical Note details the methodologies which underpin the Static Traffic Assignment (STA) component of the Zenith model. The STA model is the standard method for assigning traffic within the Zenith four-step model. A dynamic traffic assignment (DTA) method, known as Streamline, is also available and is used in custom applications.

The remainder of this document is structured as follows:

- Section 2 describes the objectives of STA
- Section 3 describes the model inputs
- Section 4 describes the modelling process
- Section 5 describes the model outputs
- Section 6 describes the limitations of the model

2 Objectives and Context

The aim of the STA model is to realistically estimate the *routes chosen* and *travel costs incurred* by motorists as they travel within the modelled area. The model is sensitive to the effects of travel time, traffic congestion and tolls, and separately models the travel made by car and freight vehicles.

Within the overall Zenith model process (illustrated in Figure 1 below), assignment is the final step. However, the particular role played by assignment is more clearly illustrated in Figure 2, which groups the model's numerous components into two: the demand model and the assignment model.

The demand model, which encompasses all of the model's components *except assignment*, is concerned with estimating the level of demand (number of trips) between each origin / destination (OD) pair, by the various trip purposes, travel modes and periods of the day.

The assignment model estimates the route chosen for each trip output from the demand model, and outputs travel costs (travel time, toll, fuel, fare, etc), which are subsequently fed back to the demand model (leading to an iterative process between the demand and assignment models).

Within the assignment model, the STA is responsible for the assignment of car and freight vehicle trips only. The assignment of public transport passengers is handled by the Transit Assignment, and takes place after the STA (to allow traffic related congestion to impact upon bus travel times and passenger route choices). Assignment of pedestrians and cyclists is also performed separately.

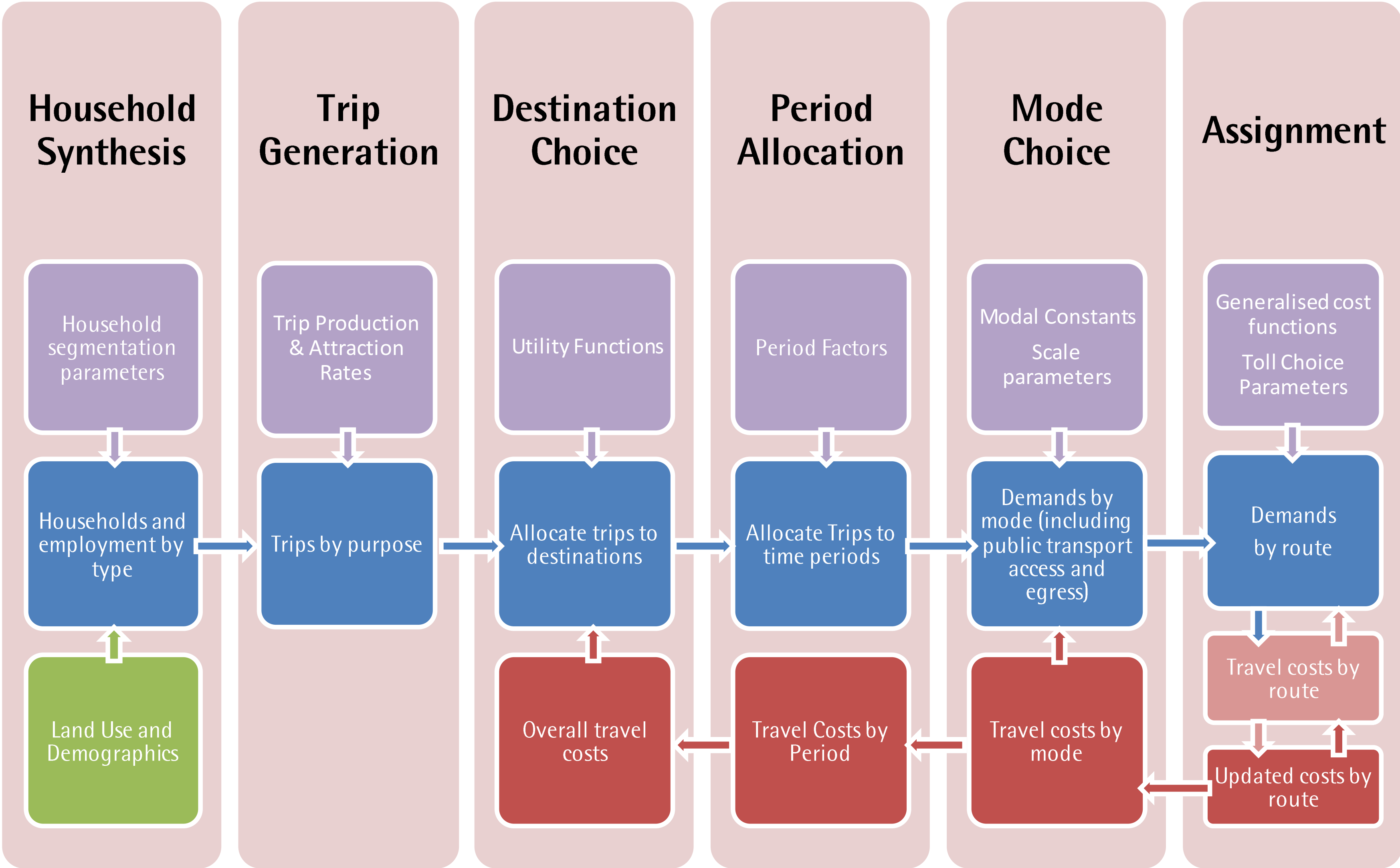


Figure 1 – Zenith Model Process

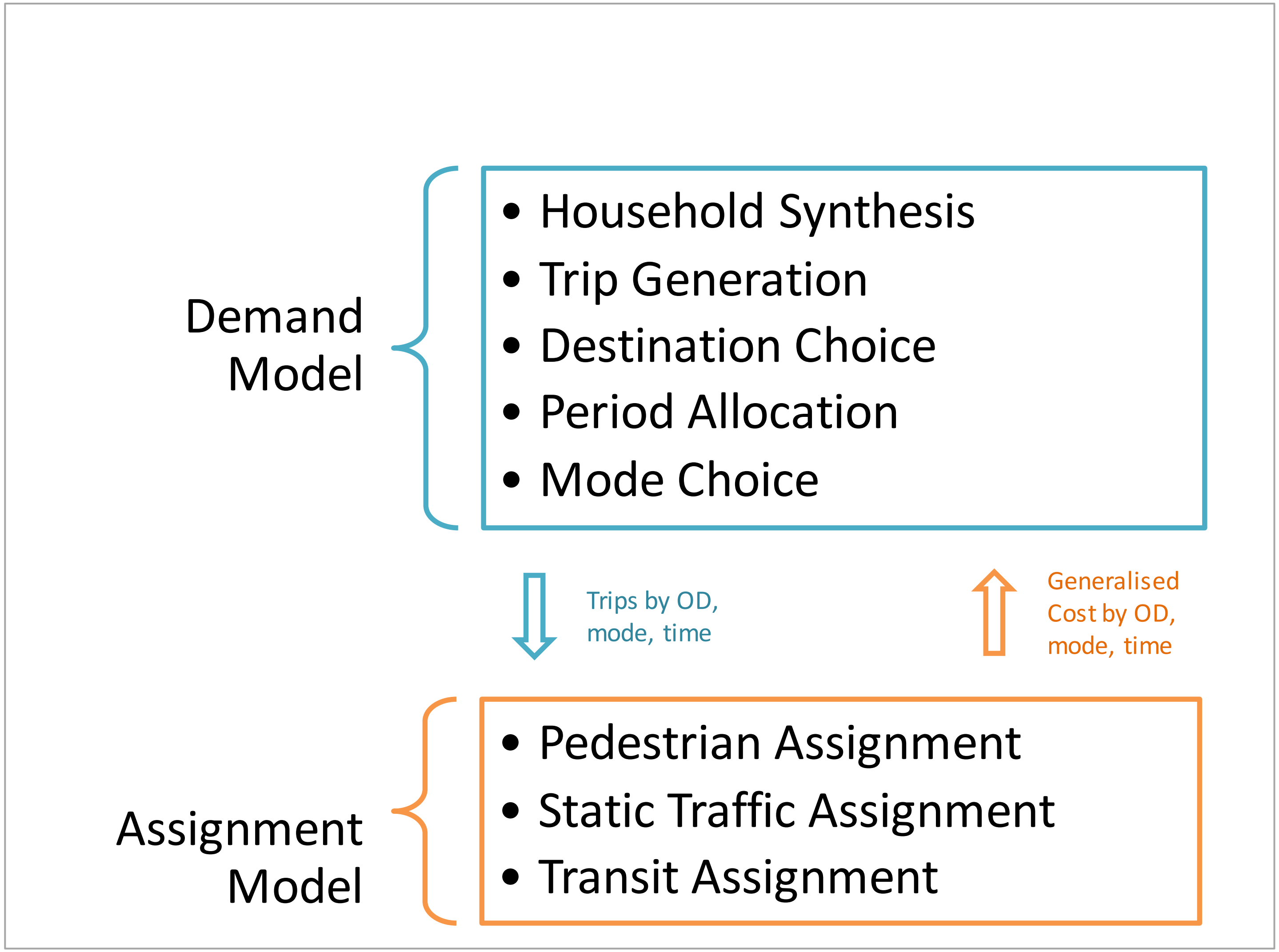


Figure 2 – The Steps of the Zenith Model can be grouped into "Demand" and "Assignment" steps

3 Inputs

This section describes the inputs to the Zenith Static Traffic Assignment (STA) model.

Road Network Attributes

- **Link type** – link types are used to apply mode restrictions to specific links. This feature could be used to ban heavy vehicles from roads where height restrictions apply or pedestrians and cyclists from freeways, for example (though we don't currently use it in this way). The link type also distinguishes freeway links from non-freeway links, which is important in the calculation of traffic speeds under congestion
- **Link free-flow speeds** – free-flow speed is defined as the expected travel speed in un-congested conditions (e.g. at 3am). For freeways, the free-flow speed is generally equal to the speed limit. For non-freeway roads, the free-flow speed is generally lower than the speed limit, due to the effect of junction controls (i.e. traffic signals, priority junctions, roundabouts) and the environment through which the road passes (i.e. on-street parking, pedestrian crossings, shopping precincts, etc.)
- **Link capacities –** link capacities are expressed in vehicles per hour (not PCUs per hour, though we may move to this definition in the future)
- **Turn restrictions –** turn restrictions are definable by mode and by time of day
- **Tolls and toll caps –** tolls and toll caps are definable by mode and by time of day

It should be evident from the above list that the Zenith STA is a *link-based* model (not a *junction-based* model). Delays resulting from congestion are reflected in reduced link speeds; not delays at junctions.

Behavioural Parameters

- **Toll choice parameters** – these parameters are input to the toll choice model, which estimates route probabilities given a set of alternative tolled and untolled routes. The derivation of these parameters is described in the Technical Note: *"Static Traffic Assignment Model – Estimation, Calibration and Validation"*.

Traffic Demand

- **Demand OD Matrices** - The number of vehicle trips travelling between each origin / destination pair is supplied as a fixed input to the STA process. Vehicle trips are stored in OD Matrices, with a separate matrix defined for each mode segment (Airport travel, Company Car, Non-Company Car, Light Commercial Vehicles and Heavy Commercial Vehicles).

Parameters Related to the Estimation of Delay due to Traffic Congestion

- **Speed-flow relationships –** the effects of traffic congestion are simulated through "speed-flow curves" which express the functional relationship between traffic volume and average traffic speed. Zenith employs two speed-flow curves, one for freeway roads, and one for non-freeway roads, as shown in Figure 3 below. The speed-flow curves take the link "volume-capacity ratio" as input (i.e. link traffic volume / link capacity), and output speed as a percentage of the free-flow speed.

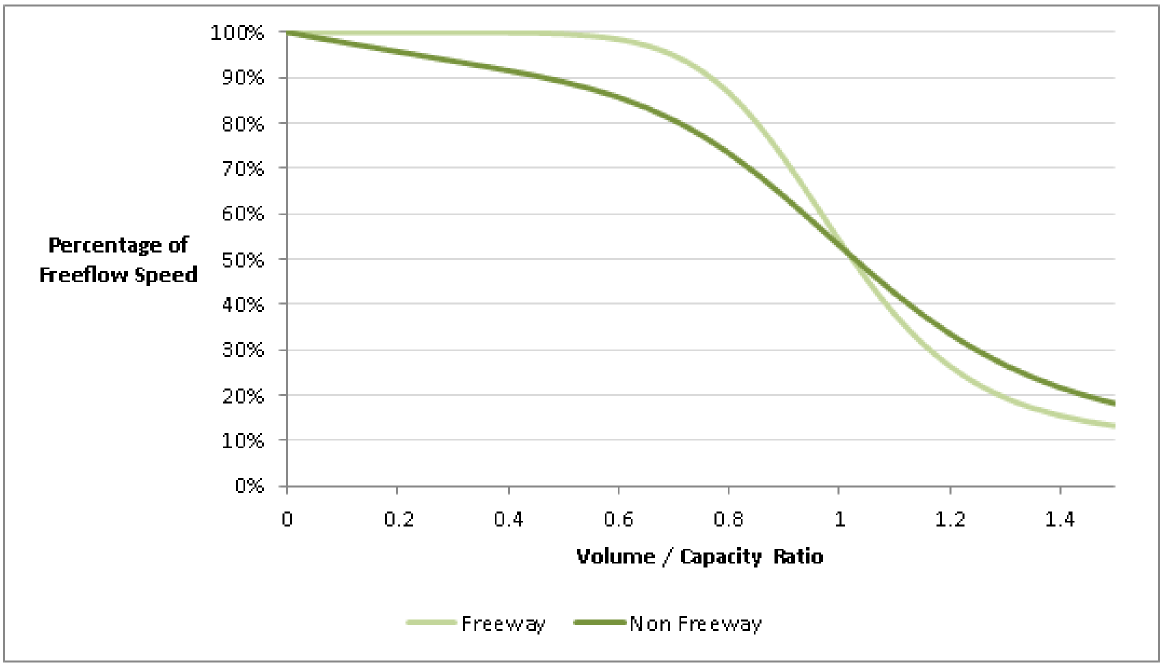


Figure 3 - Speed-Flow Curves

- **Peak Hour Factors** the Demand OD Matrix includes all trips made during a given period. Where the period is longer (or shorter) than 1 hour, a "peak hour factor" is used to scale demands to hourly flows. This allows 1 hour volume-capacity ratios to be calculated, using the previously listed hourly capacities.

<u>*Algorithmic parameters*</u>

- **Number of Iterations –** the Zenith STA employs an iterative process, which allows traffic volumes (and congestion related delays) to reach equilibrium. The number of iterations is fixed by the user, to facilitate "like-for-like" comparisons between scenarios.

4 Process

This section describes the Zenith Static Traffic Assignment (STA) algorithm.

4.1 Overview

The flow diagram presented in Figure 4 outlines the various inputs, outputs and processes which comprise the overall algorithm. Each of the model processes is described further in the sub-sections that follow.

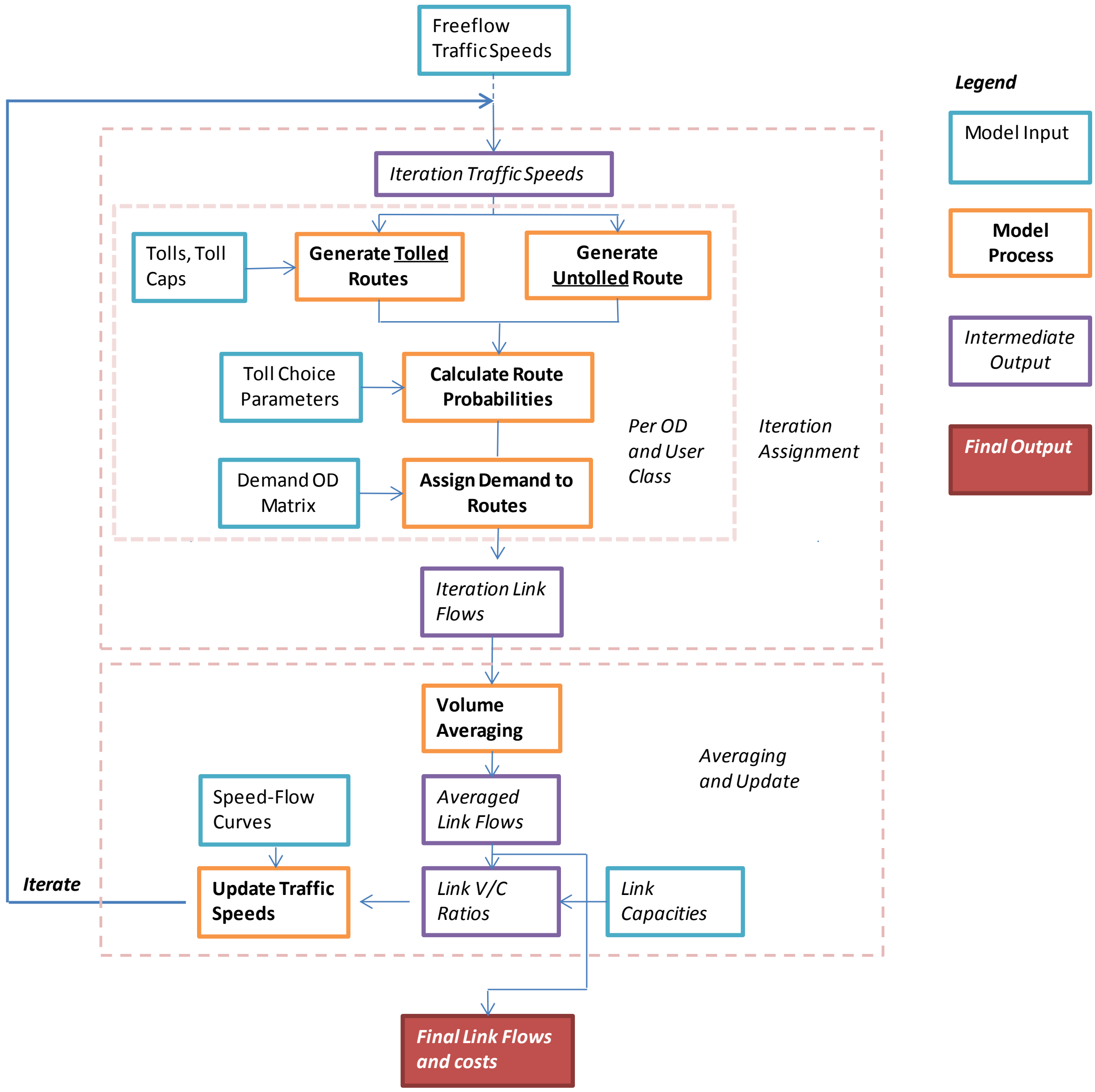


Figure 4 – The Zenith Static Traffic Assignment Process

The algorithm is necessarily iterative, to account for the two-way relationship between route choices and traffic speeds. Traffic speeds determine the travel time for each route, and thus determine driver route choices. The aggregate route choices of all motorists in turn determine link traffic flows which in turn determine congestion levels and traffic speeds.

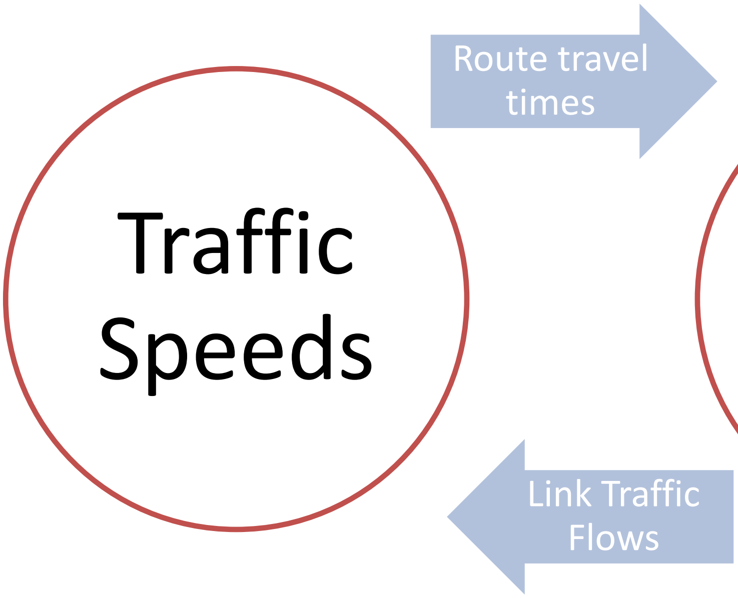


Figure 5 - The Two-Way Relationship between Traffic Speeds and Route Choices

The algorithm is executed for a fixed, user-specified number of iterations, with the user responsible for choosing an iteration count that achieves a satisfactory level of convergence. Convergence is discussed further in Section 4.8.

Each iteration is comprised of two major steps, which are themselves comprised of a series of minor steps:

1. **Iteration Assignment** – a new set of *iteration link flows* is created. This involves:
    a. **Calculating tolled and untolled routes -** based on the traffic speeds which are input to the iteration. In the first iteration, the traffic speeds are set equal to free-flow speeds (to seed the process). In all subsequent iterations, the traffic speeds are calculated based on the levels of congestion output from the previous iteration.
    b. **Calculating route probabilities -** based on the relative travel time and toll level of competing routes.
    c. **Assigning demands to routes -** all trips, for all user classes, are assigned to the network using the calculated route probabilities. The result is a set of link flows specific to the iteration, referred to as *iteration link flows.*
2. **Averaging and Update**
    a. **Volume Averaging** - the iteration link flows are averaged with the iteration link flows from all prior iterations. All iterations are weighted equally (i.e. after 10 iterations, each iteration will contribute 10% to the *averaged link flows*.
    b. **Update Traffic Speeds** – volume / capacity ratios (calculated using the *averaged link flows*) are then used to update link traffic speeds, using the speed-flow curves illustrated in Figure 3.

The final output of the algorithm is a set of *averaged link flows*, where each set of *iteration link flows* is weighted equally in the final result. The algorithm also outputs a set of *link traffic speeds* which are calculated based on the averaged link flows.

4.2 Generation of Untolled Routes

The Zenith STA algorithm generates a single untolled route for each origin / destination (OD) pair and user class. This process is performed separately per iteration, reflecting the variation in traffic speeds which occurs between iterations.

The untolled route is calculated as the shortest path (in terms of travel time), between origin and destination, avoiding all toll points.

4.3 Generation of Tolled Routes

In situations where motorists face a choice between tolled and untolled routes, the Zenith STA algorithm will generate a set of feasible tolled routes, with each route offering a different trade-off between travel time and toll.

Tolled routes are calculated separately for each origin / destination pair, user class, and iteration. A tolled route is defined as:

The shortest path between a given origin / destination pair, which traverses a specific ordered set of tolled points.

In this context, the shortest path is defined as the shortest *travel time* path, and a toll point is the location where a fixed toll is charged (i.e. a toll gantry).

For example, in a network with 3 toll locations, labelled A, B and C, there are 15 unique tolled routes:

Routes using exactly 1 toll point:

- Origin => Toll A => Destination
- Origin => Toll B => Destination
- Origin => Toll C => Destination

Routes using exactly 2 toll points:

- Origin => Toll A => Toll B => Destination
- Origin => Toll A => Toll C => Destination
- Origin => Toll B => Toll A => Destination
- Origin => Toll B => Toll C => Destination
- Origin => Toll C => Toll A => Destination
- Origin => Toll C => Toll B => Destination

Routes using exactly 3 toll points:

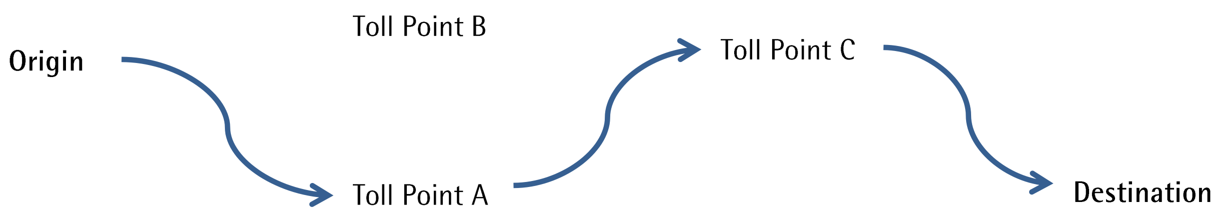
- Origin => Toll A => Toll B => Toll C => Destination
- Origin => Toll A => Toll C => Toll B => Destination
- Origin => Toll B => Toll A => Toll C => Destination
- Origin => Toll B => Toll C => Toll A => Destination
- Origin => Toll C => Toll A => Toll B => Destination
- Origin => Toll C => Toll B => Toll A => Destination

The route "Origin => Toll A => Toll B => Destination" is defined as the shortest route between origin and destination which first passes through Toll Point A and then Toll Point B.

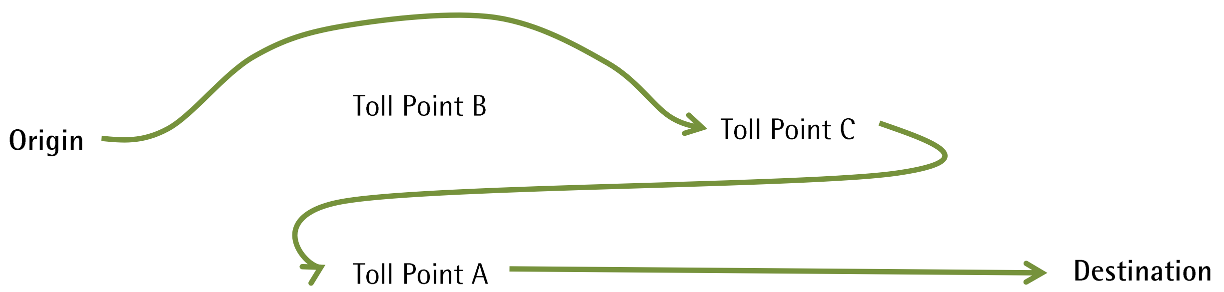
In a network consisting of 40 toll points (which is not uncommon), there are 2.2×10^{48} unique tolled routes; an impractically large number. The key, therefore, is to select a much smaller set of routes which are most attractive.

Intuitively, it is fairly easy to distinguish between "good" and "bad" routes. In the diagrams below, we first show an intuitive, potentially rational route which passes through Toll Point A and then Toll Point C, and then show an intuitively irrational route, which passes through Toll Point C, then Toll Point A.

An Intuitively Rational Tolled Route



An Intuitively Irrational Tolled Route



Algorithmically, we distinguish between good and bad routes by applying a dominance criterion, with bad routes removed from the choice set.

Dominance Criterion

A tolled route is considered irrational, and is removed from the set of tolled routes, if:

There exists another tolled route which has:

- *a lower toll; and*
- *a lower travel time*

The dominance criterion is intuitively appealing. Any user with a positive value of time should prefer a route which offers lower tolls and lower travel time. We are, of course, making the assumption that all users have a positive value of time, and that user perceptions of travel time exactly match with the model's estimates of travel time. While not strictly correct, these simplifying assumptions are also made when performing traffic route choice in the *absence* of tolls. Therefore, the assumptions are consistently applied irrespective of the presence of tolls.

When applying the dominance criterion, the untolled option is also included, but with a toll of zero. This has the effect of removing all tolled options which are slower than the untolled route. In other words, all tolled routes must provide a time saving over the untolled route. For many ODs, this results in the set of tolled routes being empty.

After applying the dominance criterion, the set of feasible tolled options has the property that when sorted in order of *ascending travel time*, the tolled routes naturally sort themselves in order of *descending toll*.

Toll Caps

By defining tolled routes in terms of an ordered list of toll points visited, it is straight forward to apply toll caps and other adjustments. Once the total toll for a given toll road exceeds the toll cap, the total toll for that toll road is simply set equal to the toll cap. Separate toll caps can be defined for separate toll roads (where a toll road is defined as a set of toll points).

User Classes

A separate set of tolled routes is constructed for each user class (e.g. car, commercial vehicle). This is necessary for two reasons:

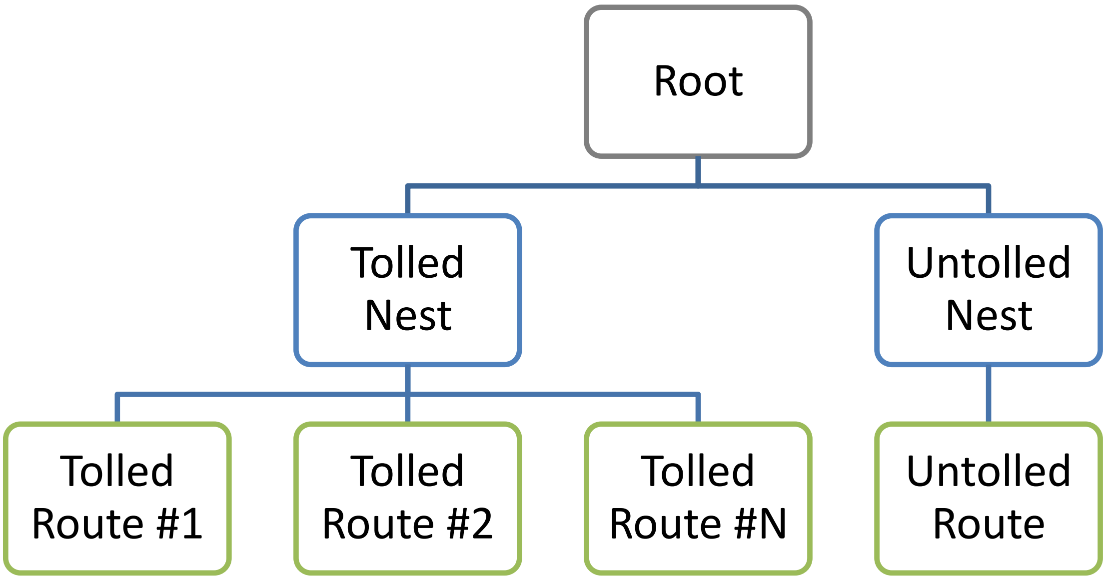
- Network attributes such as link type, speed, etc, can vary between user classes, resulting in different paths between toll points, and different travel times; and
- Toll levels and toll caps often vary between user classes.

4.4 Calculation of Route Choice Probabilities

Following the generation of tolled and untolled routes, the next step is to calculate route choice probabilities.

For each road user, travelling between a specific origin / destination pair, and belonging to a specific user class, the model will have generated a set of potential tolled routes, and a single untolled route.

The calculation of route probabilities is performed using a *nested logit model*. The set of tolled routes are organised into a tolled nest, with the untolled nest consisting of the single untolled route. The structure is shown in the diagram below.



The calculation of route probabilities then follows a two-step process:

1. Calculate probabilities for the binary choice: toll versus no toll (a choice between the Tolled Nest and the Untolled Nest).
2. Divide the total probability of the Tolled Nest among each of the tolled routes.

Mathematically, both steps can be treated as a multi-nomial logit model, with information from lower levels of the model (ie. the tolled routes) informing the utilities of the upper level nests (ie. the tolled nest).

Each step will now be examined in more detail.

4.4.1 Calculation of Probabilities – Tolled versus Untolled

At the upper level of the nested logit model, the choice between tolled and untolled routes is treated as a binary choice (between the tolled nest and the untolled nest).

Expressed as a binary logit model, the probability that a tolled route will be chosen is calculated as follows:

$$P(tolled) = \frac{1}{1 + e^{(V_{untolled} - V_{tolled})}}$$

Where,

- *P(tolled)* is the probability assigned to the tolled nest
- $V_{untolled}$ is the estimated mean utility of the untolled nest, and
- V_{tolled} is the estimated mean utility of the tolled nest.

The calculation of estimated mean utilities, $V_{untolled}$ and V_{tolled} is relatively straightforward, with both utilities calculated as a linear function of travel time and toll (using identical parameters), as follows:

$$V(route) = \beta^{U}_{time} \times time_{route} + \beta^{U}_{toll} \times toll_{route}$$

Where,

- *V(route)* is the estimated mean utility of a given route
- $time_{route}$ is the travel time of the route (in minutes)
- $toll_{route}$ is the total toll payable (in cents) for the route (after applying toll caps, etc)
- β^{U}_{time} and β^{U}_{toll} are model parameters, which can vary by user class and user segment. The *U* indicates that the parameters are specific to the "upper level" choice between the tolled nest and the untolled nest.

When calculating $V_{untolled}$, the values of $time_{route}$ and $toll_{route}$ are based on the single untolled route; $toll_{route}$ will be zero by definition!

When calculating V_{tolled}, the values of $time_{route}$ and $toll_{route}$ are drawn from the *most attractive tolled route* (the tolled route with highest utility).

Expanding out the calculations of *V*, the tolled probability can also be expressed as follows:

$$P(tolled) = \frac{1}{1 + e^{(\beta^{U}_{time} \times TS - \beta^{U}_{toll} \times toll)}}$$

Where,

- *TS* is the 'time saved' by the most attractive tolled route (compared with the untolled route)
- *Toll* is the 'toll payable' on the most attractive tolled route

Worked Example

If we assume that:

- the untolled route has a travel time of 30 minutes, and
- the most attractive tolled route has a travel time of 20 minutes (a time saving of 10 minutes), and has a toll of $3 (or 300 cents), and
- β^U_{time} = -0.3 and β^U_{toll} = -0.008

Then,

$$P(tolled) = \frac{1}{1 + e^{(-0.3 \times 10 - -0.008 \times 300)}}$$

$$P(tolled) = 0.646$$

In this case, the estimated tolled probability (the probability that a tolled option will be chosen) is 64.6%. The untolled probability is 35.4%.

By solving for different toll levels, we can plot what is often referred to as a toll diversion curve, which traces out the value of $P(tolled)$ for varying toll levels (and fixed time saving). In Figure 6 below we plot a toll diversion curve using the example parameters listed above, and a fixed time saving of 10 minutes. Note that these parameters (and the resulting curve shown below) are not real, they are illustrative only.

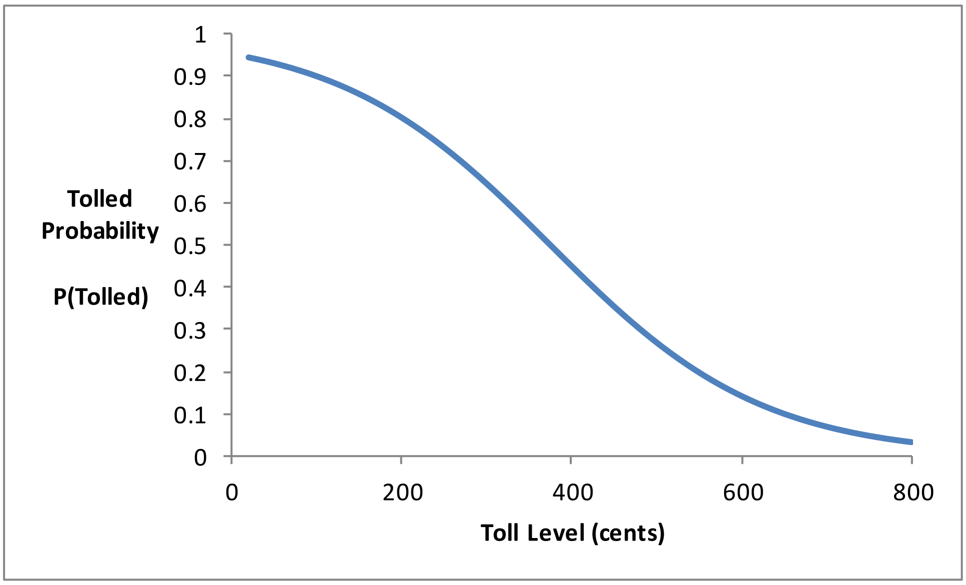


Figure 6 - A toll diversion curve, plotting tolled probability as a function of toll, for a fixed time saving of 10 minutes

4.4.2 Calculation of Probabilities – Allocation between Tolled Routes

Having calculated the tolled probability, the second step is to allocate probabilities to the individual tolled routes.

This choice can again be expressed a multi-nomial logit, with the estimated mean utility of each tolled route calculated as follows.

$$V_r = \beta_{time}^{L} \times time_r + \beta_{toll}^{L} \times toll_r$$

Where,

- V_r is the utility of tolled route r
- $time_r$ is the travel time of tolled route r
- $toll_r$ is the toll payable for tolled route r
- β_{time}^{L} and β_{toll}^{L} are model parameters, with the L referring to the fact that these parameters are specific to the "lower level" choice between tolled alternatives.

Once all the estimated mean utilities are calculated, the probability that route r will be chosen, (conditioned on a tolled route being chosen) can be calculated as follows:

$$P_r = \frac{e^{V_r}}{\sum_j e^{V_j}}$$

Therefore, the unconditioned, absolute probability that tolled route r will be chosen is:

$$P_r = P(tolled) \times \frac{e^{V_r}}{\sum_j e^{V_j}}$$

It is worth noting the relationship between the lower level model parameters β_{time}^{L} and β_{toll}^{L}, and the upper level model parameters β_{time}^{U} and β_{toll}^{U}. Generally speaking, β_{time}^{L} and β_{time}^{U} are set equal. However, it is common to set β_{toll}^{L} to be smaller (in absolute terms) than β_{toll}^{U}.

Intuitively, this lowers the perceived cost of tolls in the lower level choice between tolled routes (compared with the upper level choice between toll / no toll). This is motivated by the fact that the lower level model (tolled route versus tolled route) is only applied to tolled users, and as a group, tolled users have a higher value of time.

It is also worth emphasising that these parameters vary between user segments. The typical user segments within the Zenith STA are:

- Airport travel
- Travel in a company car (for any trip purpose other than airport)
- Travel in a non-company car (for any trip purpose other than airport)
- Light commercial vehicles
- Heavy commercial vehicles

4.5 Assignment of Demands to Routes

Following the calculation of route probabilities, demands are assigned onto the road network.

Demands are input in the form of demand OD matrices; one per user class. The demand for each OD is allocated to the available routes (tolled and untolled) in accordance with the calculated route probabilities.

4.6 Volume Averaging

Following the assignment of demands for all ODs and all user classes, a full set of *iteration link flows* is recorded. This set of link flows is specific to the just completed iteration, with similar sets of link flows available for all prior iterations.

The Volume Averaging process averages all sets of iteration link flows, link-by-link, to produce a single set of *averaged link flows*. Generally, each iteration is weighted equally, so that after N iterations, the averaged link flows will be the average of the iteration link flows from iterations 1, 2, …, N.

As a result, the final link flows output from the assignment will be the average of the iteration link flows from all of the iterations completed.

4.7 Update of Traffic Speeds

At the conclusion of each iteration, following the calculation of *averaged link flows* (as per Section 4.6), a new set of link traffic speeds is calculated, to be used as input to the following iteration.

The traffic speed of each link is calculated independently, based on its volume / capacity ratio. The volume input to the volume / capacity ratio is calculated by summing the averaged link flows across all user classes.

Two speed-flow curves are used, as shown in Figure 3.

4.8 Convergence

Run for an infinite number of iterations, the Zenith STA algorithm will eventually converge to a perfectly stable solution. However, with only a finite number of iterations at our disposal, it is important to monitor the level of convergence achieved.

Although there are several alternatives, the key metric used to monitor Zenith STA convergence is the relative changes of link costs between consecutive iterations, known as the relative gap, or RGAP, where:

$$RGAP = \frac{\sum_{a=1}^{N}\left(V_a^{n-1}C_a^{n-1}\right) - \sum_{a=1}^{N}\left(F_a^{n}C_a^{n-1}\right)}{\sum_{a=1}^{N}\left(V_a^{n-1}C_a^{n-1}\right)}$$

where

- C_a^{n-1} is the generalised cost of link a at the conclusion of iteration n-1 (which is also the cost input to iteration n)
- F_a^{n} is the *iteration link flow* for link a in iteration n based on C_a^{n-1}
- V_a^{n-1} is the *averaged link flow* for link a at the conclusion of iteration n-1

Normally, VLC consider a relative gap of less than 1% acceptable. This is consistent with the VicRoads Transport Modelling Guidelines (Volume 2 – Strategic Modelling).

5 Outputs

The standard outputs of the Zenith STA are:

- Link flows, by user class
- Turning movements, by user class
- OD Skims (weighted average across tolled and untolled routes)
    - Travel time
    - Distance
    - Toll
- Toll revenue, by toll point and user class
- Convergence levels (e.g. relative gap), by user class and iteration

Additional non-standard outputs include:

- Cordon matrices
- Select link matrices, select link loads

6 Limitations

Like all models, the Zenith STA has a number of limitations. The key limitations are:

- **Junctions not explicitly modelled –** it is well known that network junctions (signalised, priority, roundabout, etc.) are often the source of congestion related delays, rather than link mid-blocks. However, the Zenith STA does not explicitly include junctions or their delays; rather, links are assigned a capacity, and have their speed reduced depending on the degree to which this capacity is consumed. To some extent, junction delays can be captured in a link based model by assigning link capacities that reflect an estimate of junction constraints (i.e. estimated signal green time). However, this approach is not as robust as explicitly coding junctions. On the other hand, the main arguments against junction modelling are related to the availability of quality junction data (for both here and now, and for the future), and added complexity. We intend to explore removing this limitation in the future.
- **Queues not modelled** – traffic speeds are calculated for each link *independent* of all others. However, in reality, queues from one link can propagate up-stream causing delays on other links. This limits the ability of the model to model traffic speeds under very heavy congestion. One solution to this limitation is to employ a dynamic traffic assignment model – VLC are currently building a dynamic traffic assignment prototype, and hope to release it in the near future.
- **Value of time is a point estimate –** within the nested logit route choice model, the parameters on time and toll are fixed point estimates (not distributions), resulting in a single value of time per user class. In reality, user perceptions of time and the value of it follow a distribution. Despite this limitation, the model is still able to model heterogeneous user preferences through the "error term" implicit in the nested logit model. In the future, we may explore the implementation of a mixed logit model, which accommodates random parameters.
- **Correlation (overlap) between tolled options not considered** – when calculating the choice between tolled routes, we employ a multi-nomial logit model, which does not account for correlation between similar tolled alternatives. This may result in an over-estimation of probabilities to clusters of very similar alternatives. It is difficult to say whether this is a major issue, as its effects have not been explored in a real setting. Approaches exist which could address this issue (e.g. the paired combinatorial logit model), and their implementation could be explored if this proves to be important in the future.