

Omnitrans International

# Junction Modelling in OmniTRANS

Technical Note - October 2007

## Document and Contact Information

### Document

Date: 8-11-2007  
Reference: Junction Modelling in OmniTRANS  
OmniTRANS version: 4.2 or later  
Author(s): Erwin Bezembinder, Feike Brandt

### How to Contact Omnitrans International

	Snipperlingsdijk 4 7417 BJ Deventer The Netherlands	Address
	+31-(0)570-666111 +31-(0)570-666888	Phone Fax
	P.O. Box 161 7400 AD Deventer The Netherlands	Mail
	info@omnitrans.nl	Email
	www.omnitrans-international.com www.omnitrans.nl	Website
	techsupport@omnitrans.nl	Technical support

### *OmniTRANS ©*

*The software described in this document is furnished under a licence agreement. The software may not be used or copied only under the terms of the licence agreement. No part of this paper may be photocopied or reproduced in any form without prior consent from Omnitrans International B.V.*

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	What is Junction Modelling in OmniTRANS?.....	5
1.2	When to Use Junction Modelling in OmniTRANS.....	6
1.3	How to Use Junction Modelling in OmniTRANS.....	6
1.4	Outline of this Document.....	7
<b>2</b>	<b>Input .....</b>	<b>8</b>
2.1	Standard Input.....	8
2.1.1	Drive Rule.....	8
2.1.2	The Junction Editor.....	9
2.1.3	Junction Type.....	11
2.1.4	Junction Attributes.....	11
2.1.5	Approach and Lane Data.....	13
2.1.6	Signal Settings.....	18
2.1.7	Check Button.....	23
2.1.8	Examples.....	24
2.2	Advanced Input.....	27
2.2.1	Junction Modelling Defaults.....	27
2.2.2	Project Defaults.....	28
2.2.3	Junction Type Defaults.....	30
2.2.4	Network Editor.....	37
2.2.5	Database.....	40
2.2.6	Upgrading Old Projects.....	41
<b>3</b>	<b>Modelling.....</b>	<b>42</b>
3.1	General Junction Modelling Properties.....	42
3.2	Junction Modelling within a Static Assignment.....	44
3.2.1	Traffic Assignment and Junction Modelling.....	44
3.2.2	All-or-Nothing Assignment.....	45
3.2.3	Incremental Assignment.....	45
3.2.4	Volume-Averaging Assignment.....	46
3.2.5	Equilibrium Assignment.....	46
3.2.6	Generalised Cost Formulation.....	46
3.2.7	Skim Generation.....	47
3.2.8	Multi-User-Class Assignment.....	47
3.3	Junction Modelling within a Dynamic Assignment.....	48
3.3.1	Dynamic Traffic Assignment and Junction Modelling.....	49
3.3.2	Static Route Choice and Junction Modelling.....	51
3.3.3	Static Route Choice and Dynamic Junction Modelling.....	52
3.3.4	Dynamic Route Choice and Junction Modelling.....	52
3.3.5	Dynamic Multi-User-Class Assignment.....	52
3.4	Error and Warning Messages.....	52
<b>4</b>	<b>Output.....</b>	<b>58</b>

4.1	Junction Data .....	58
4.1.1	Critical v/c ratio.....	60
4.1.2	Weighed average v/c ratio .....	61
4.1.3	Average delay .....	61
4.1.4	Average queue.....	61
4.1.5	Level of service .....	62
4.1.6	Calculated cycle time .....	62
4.2	Turn Data .....	62
4.2.1	Load .....	64
4.2.2	Capacity .....	64
4.2.3	Cost.....	65
4.2.4	Queue .....	65
4.2.5	Calculated Green Time .....	65
4.2.6	Calculated Green Time Off Set .....	65
4.3	Signal Data .....	65
4.4	Lane and Link Data for Dynamic Assignment .....	66
4.5	Database .....	68
<b>5</b>	<b>Methodology .....</b>	<b>69</b>
5.1	Symbols .....	69
5.2	Equal junction .....	69
5.2.1	Capacity.....	70
5.2.2	Delay .....	72
5.3	Priority junction .....	73
5.3.1	Capacity.....	74
5.3.2	Delay .....	78
5.4	Signalized junction .....	79
5.4.1	Capacity.....	79
5.4.2	Delay .....	82
5.4.3	Calculation maximum cycle time .....	84
5.5	Unsignalized roundabout.....	86
5.5.1	Capacity.....	86
5.5.2	Delay .....	91
<b>6</b>	<b>Average Overflow Queue.....</b>	<b>92</b>
<b>7</b>	<b>References .....</b>	<b>93</b>

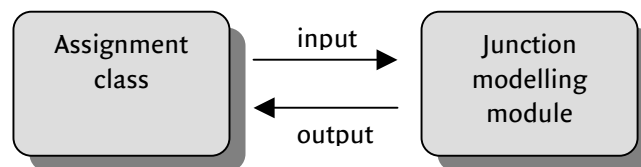
# 1 Introduction

OmniTRANS incorporates a powerful junction modelling module which can be used within both the static and the dynamic traffic assignment modelling classes, respectively OtTraffic and OtMadam. This document provides detailed information about the use and background of this junction modelling module.

## 1.1 What is Junction Modelling in OmniTRANS?

**The main objective of junction modelling is to calculate the average delay per vehicle for each turning movement on the basis of the junction layout, turning flows and optionally signal settings. The turning delays are then used in the route choice and blocking-back processes of the assignment model.**

As well as turning delays, the junction modelling module also stores the calculated turning capacities, green times and queue lengths as well as a series of performance measures for the whole junction, such as the volume/capacity ratio, the average delay, the average queue and a level of service value.



*Figure 1 - Junction modelling module communicating with an assignment class.*

Junction modelling can model:

- Signalised junctions and roundabouts;
- Uncontrolled junctions (no signs and/or signals);
- Sign-controlled junctions (two-way stop, all-way stop and give-way/yield);
- (Mini-)roundabouts.

For all junctions, the maximum allowed arms is four.

Freeway merges, weaving segments and ramp junctions are modelled separately in OtMadam.

The junction modelling module enables you to:

- Select automated, pretimed (fixed-time) or vehicle-actuated signal operation for each junction separately;
- Include the effects of pedestrian and/or bicycle crossings;
- Define any number and combination of approach lanes;
- Use automated calculation of optimal cycle time and green times;
- Handle normal and two-stage gap acceptance for opposing flows;
- Model approach flaring, short and/or slip lanes;
- Allow for complex lane sharing and utilisation;
- Perform iterative simulation of over-saturated roundabouts.

## 1.2 When to Use Junction Modelling in OmniTRANS

Junction modelling is generally considered as being an option for when junction delays are a significant component of the travel cost, often in peak period car assignment models. In OmniTRANS, the assignment model can be either static or dynamic. The static models are normally using capacity restraint techniques to deal with congestion. Within this context, the use of junction modelling is advisable when one wants to model:

- Congested urban areas with a significant number of queues and/or conflicting flows at junctions;
- The effects of modelling a different Junction Type for a given junction;
- Signal settings strategies;
- Dynamic traffic management measures;
- Accurate door-to-door or route segment travel times;
- Departure time choice modelling or other travel time sensitive feedbacks.

Junction modelling in OmniTRANS is designed to determine the turning delays for motorised traffic. Although it can take account of the presence of pedestrian and/or bicycle crossings and bus priority lanes, it does not determine delays or any other performance measures for these modes of transport.

Junction modelling can also be used in combination with a static or dynamic multi-user-class assignment.

## 1.3 How to Use Junction Modelling in OmniTRANS

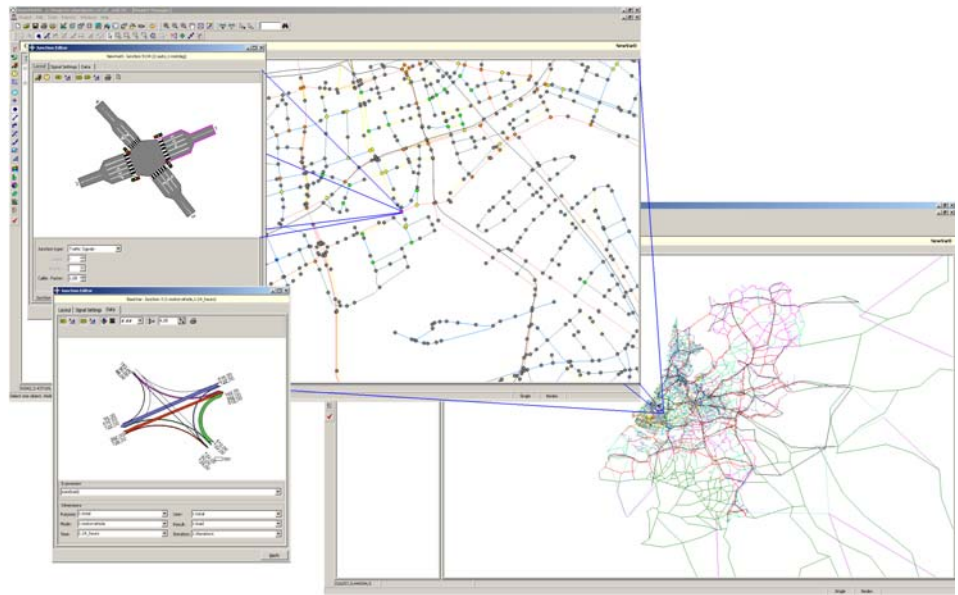
Given a link based network, nodes that are to be modelled as junctions must be coded appropriately. To invoke the junction modelling the appropriate junction modelling property in the relevant assignment class (OtTraffic or OtMadam) is enabled.

In an OmniTRANS network any node that has three or more connected links can be defined as modelled junction by giving it a 'junction definition'. This can be added using the Junction Editor, or is done automatically if the network is imported using an import class that supports junctions. Once a node is given a junction definition, i.e. it has been assigned a basic Junction Type, a series of layout attributes and optionally some signal settings, junction modelling can be used to calculate delays for each turning movement at the junction.

Generally, two levels of detail are used in an assignment model. The relevant 'inner' study area is coded in more detail than the surrounding 'outer' area<sup>1</sup>. In most models, the junctions in the 'inner' area are defined as such. In the 'outer' area, no junction definitions are used. In some models, default turn penalties are used in the latter case. It is up to the user to decide which levels of detail are used in the different parts of the network. The 'inner' area should have sufficient coverage such that extra delays calculated at junctions are not so great that traffic seeks 'cheaper' by-pass routes through the 'outer' network.

---

<sup>1</sup> SATURN users will recognise the distinction between the simulation and the buffer network.



*Figure 2- Junction modelling in OmniTRANS.*

## 1.4 Outline of this Document

The remaining Chapters will provide more detailed information about the topics discussed in this introduction.

The second Chapter gives a detailed description of the input data and provides a series of practical tips for coding junctions. The third Chapter explains how junction modelling can be used within assignment modelling classes. The fourth Chapter lists the available output data and suggests various ways to present and analyse the data. The fifth Chapter briefly describes the most important equations from the methodology. The document ends with a list of references which have been used for the methodology.

## 2 Input

This Chapter gives detailed information on the input data for junction modelling. For practical reasons, the Chapter is divided in two main sections, respectively for standard and for advanced input.

The standard section covers the information about the junction input that is sufficient for most users of OmniTRANS. This section explains how you can code junctions using the Junction Editor. The Junction Editor enables you to use predefined Junction Types, including a set of default values for the modelling of these junctions, and lets you code a series of general variables to define the layout and the signal settings of the junction. The section also offers a reasonable number of examples. Everything that is explained in this section is implemented in the Junction Editor.

The advanced section goes into more detail about changing default values for both general behaviour on junctions as well as the default values for each predefined Junction Type. It also explains how you can use the Network Editor as an alternative for the Junction Editor and shows the database tables and fields that are used to store the junction information. Some of the items explained in this section are controlled by editing control files outside OmniTRANS.

### 2.1 Standard Input

The standard, fastest and most convenient way of coding junctions in OmniTRANS is by using the Junction Editor. This will be explained extensively in the following paragraphs, including numerous examples.

#### 2.1.1 Drive Rule

When you start a new project in OmniTRANS, you will have to define the drive rule. This option is part of the so-called project preferences. The project preferences can be controlled by way of the Preferences window (Ctrl+F). The drive rule option can be found on the Network tab of the Preference window and is called 'Left hand rule of road'. When this option is 'checked' the left-hand driving rules apply for all networks in the project, otherwise the right-hand driving rules apply.

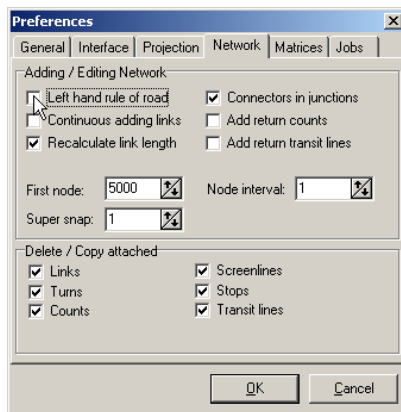
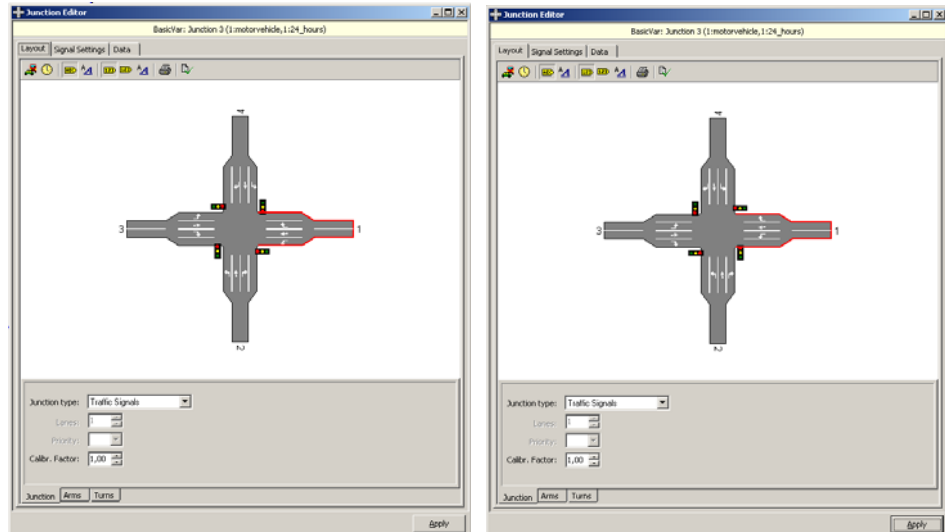


Figure 3 - Setting the drive rule in the Preferences window.

The option is inaccessible if a Network window (or tab) is open. Close all Network windows (or tabs) before activating the Preferences window to make it accessible.

Generally, the drive rule is set at the start of a project. However, OmniTRANS permits you to switch the rule at any stage. The Junction Editor automatically puts the already defined approach lanes on the chosen side of the road.



*Figure 4 - Left- and right-hand drive rules in the Junction Editor.*

However, it is your responsibility how to deal with the direction of one-way links in the network and thus how these approach junctions. The direction of the one-way links will NOT change when you change the drive rule of the project. A quick way to change the direction of all the links in the network would be to select them all (Ctrl+A), open the Attribute Editor window (F12) and press the 'Swap link directions' button.

### 2.1.2 The Junction Editor

The Junction Editor (Ctrl+J) is used for editing and analysing junctions in OmniTRANS.

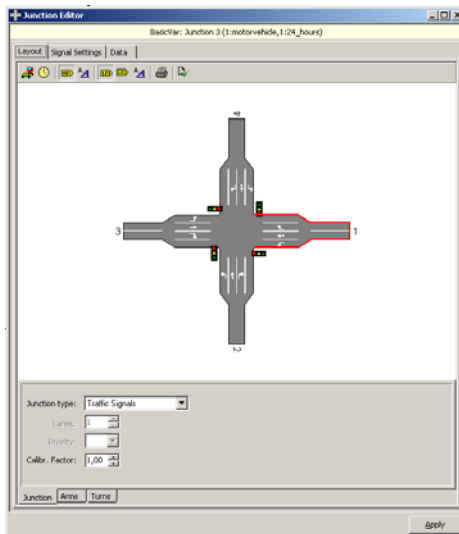


Figure 5 - Junction Editor layout tab.

Junctions are connected to OmniTRANS node objects. To edit a junction, select a node in the Network Editor and open the Junction Editor. The Junction Editor is a so-called 'on-top' window, it stays visible when another node is selected in the network. You can also type a number in the Node textbox and press enter to 'jump' to a specific junction (node) or use the PgUp/PgDn keys in that textbox to scroll through the junctions (nodes) in the network. For a full description of the Junction Editor see the OmniTRANS manual.

Depending on the database structure of the project, the Junction Editor has two or three tabs. Projects that have been created with a template from an OmniTRANS version pre 4.2 only have a Layout and a Loads tab. Projects that have been upgraded or have been based on a template from version 4.2 or later have an additional Signal Settings tab. For an explanation of how to upgrade an old project, see § 2.2.6.

This Chapter only describes the two input related tabs of the Junction Editor; Layout and Signal Settings. The Loads tab will be described in Chapter 4.

Note that both the Layout and Signal Settings tab have four buttons at the bottom of the window. The Apply button is the most important one, since any changes made in the Junction Editor for a particular node are not stored in the database of the project unless the Apply button is pressed. The Clear button removes all Layout and Signal Settings immediately (also from the database). The Check button is very useful when you want to check whether your junction configuration is correct. It performs a check and reports possible errors. The Print button will be explained in Chapter 4.

When defining a junction, you need to define:

- What Junction Type it is;
- The characteristics of the 'central' area of the junction;
- The characteristics of each arm of the junction, both entry and exit;

Signal settings can be defined optionally.

### 2.1.3 Junction Type

If you want to model a particular junction, i.e. reckon with the delays for each turning movement, the node has to be given a particular Junction Type. The Junction Type can be chosen on the Layout tab of the Junction Editor.

You have the following options:

- 0. <undefined>
- 1. Equal (or uncontrolled) junction;
- 2. Give way (or two-way yield sign controlled, or priority) junction;
- 3. Signalised junction;
- 4. Roundabout;
- 5. Signalised roundabout;
- 6. All-way-stop controlled junction.

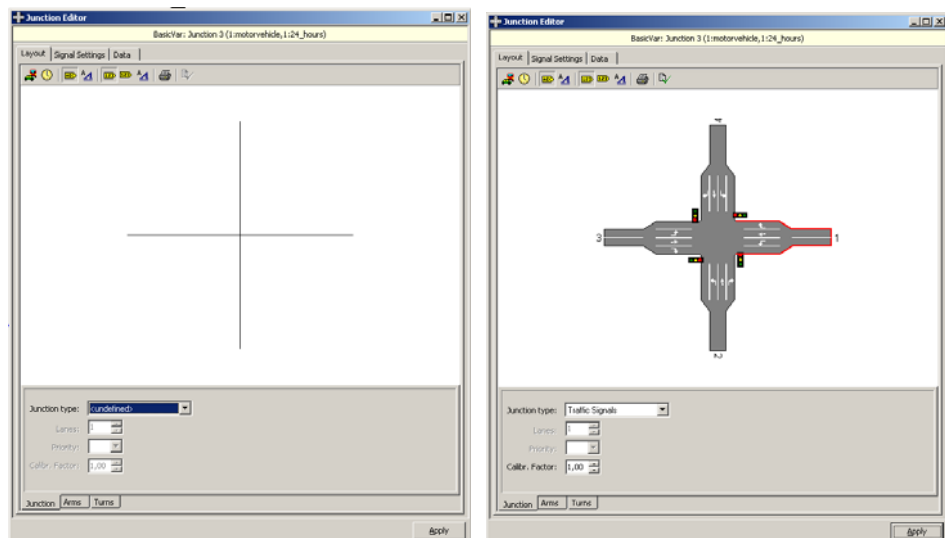


Figure 6 - Undefined and defined junction.

Junction Type '<undefined>' (see Figure 6, left), means that there is no junction definition for this node and therefore no junction modelling calculations will be performed for it. Thus, junctions in the network which are not to be modelled should be left as '<undefined>'.

The Junction Types in the Junction Editor are directly linked with the Junction Types used to gain the default values (see § 2.2.2). Different modelling methodology is used for different Junction Types (see Chapter 5).

Depending on the chosen Junction Type, one or more junctiontype specific attributes must be set. These are discussed below.

### 2.1.4 Junction Attributes


Given the selected the Junction Type, three other junction related attributes are available, although the availability depends upon the chosen Junction Type.

### Lanes on a roundabout (Junction Types 4,5)

Number of circulating lanes on a roundabout (default: 1) (not defined for other Junction Types). OmniTRANS allows you to define either one or two circulating lanes. Larger (gyratory) roundabouts should be modelled by decomposing the roundabout into a series of separate T-junctions (see § 2.1.8).

The number of circulating lanes influences the capacity of the roundabout, as well as the effect of semi-conflicts, which in turn influences the delay of the traffic entering the roundabout.

### Priority (Junction Types 2,6)

Priority movement through the junction. This property is only available for yield- and stop-sign controlled junctions. The Junction Editor provides a list of possible arm-to-arm combinations for the priority movement. The arm numbers can be displayed in the Junction Editor by pressing the first  button. The selected arms will have priority and either yield or stop signs will be placed on the non-selected arms. It is possible to define both 'straight' and 'turning' priorities.

For stop-sign controlled junctions there is an all-entry option, where a stop sign is placed on all arms (see Figure 7, right side).

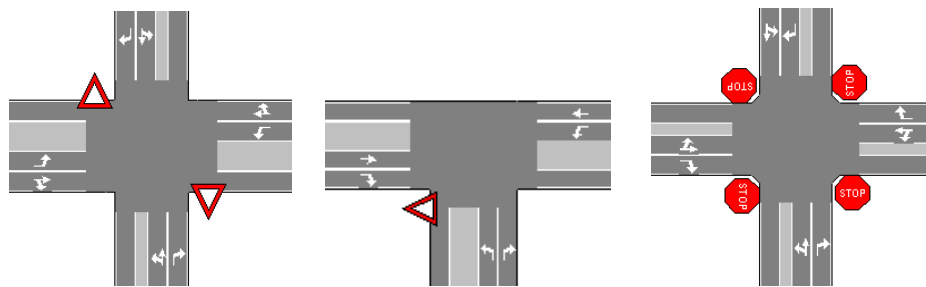


Figure 7 - Two-way yield straight, one-way yield turning, all-way stop.

### Calibr. Fac. (all Junction Types)

Calibration factor (default: 1.0). A factor applied to the calculated delay for each turn at the modelled junction and is used mainly in combination with a static assignment model. The factor is used in the following circumstances:

- Overestimation of traffic flows in static assignment models. Static models have a tendency to overestimate traffic flows on a route, since traffic is not queued in time. Although matrix calibration models should deal with this, it is sometimes desirable to have some control over the estimated delay, i.e. to increase or reduce the delay on a junction;

Besides a junctionbased calibration factor, there is also a turnbased calibration factor.

- Coordinated signals. The calibration factor can be used to reduce the delay for a set of coordinated signals. This is a very rudimentary way of dealing








with coordinated signals, since it reduces the delay for all turning movements, instead of only the intended coordinated movements.<sup>2</sup>

**2.1.5 Approach and Lane Data (all Junction Types)**

For each arm of a junction, the number and type of entry (or approach) and exit lanes must be defined. Furthermore, the width of the central reservation and the presence of crossing slow traffic can be defined. The attributes of each arm can be approached separately by selecting one of the tabs or by clicking on the arm in the Layout display area.

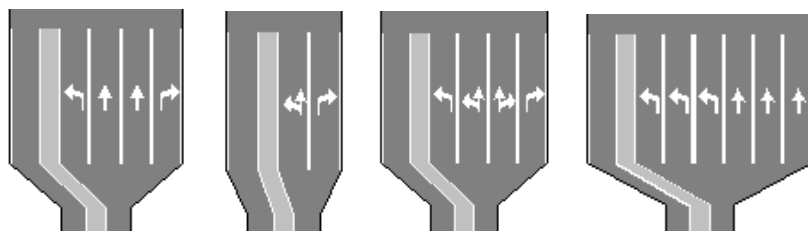
**Approach Lanes**

With the approach lane property, you can define the number and the utilisation of the entry lane types for the selected arm. Besides choosing from a predefined list of lane configurations, you can also type letters in the text box yourself. The following keys will generate the following lane types:

Key	Symbol	Description
L		Left turn lane
S		Main or straight lane
R		Right turn lane
Q		Left turn and main lane
P		Right turn and main lane
U		Left and right turn lane
A		All directions
B		Bus lane
0-9	<b>0..9</b>	Special lane (see Table 2)

*Table 1 - Lane types and associated keys and symbols.*

Figure 8 shows some examples of possible lane configurations.



*Figure 8 - Examples of lane configurations (LSSR, QR, RQPL, RRRSSS).*

Technically, there is no limit to the number of allowed lanes, however, practice will limit the number of desired options.

There are some rules to note.

*Number of lanes on a link*

<sup>2</sup> In a future version of OmniTRANS it is planned to offer the possibility to control coordination of signals by turning movement or arm. The required field is already in the database, but the modelling does not use it yet.

Although it is out of the scope of Junction Modelling, a small explanation about the number of lanes on a link is made here.

In the Junction Editor, the number of approach lanes for an a link entering a junction is different then the number of lanes on the related link.

For static assignment modelling, the number of lanes is not used. The link property 'capacity' will be used.

For dynamic modelling however, the number of lanes is important. The capacity for a link is now based on two link properties: 'lanes' and 'satflow' (saturation flow, i.e. capacity for one lane).

Because the number of lanes on an entering link is related with the number of approach lanes for a junction, the number of lanes on a link may not exceed the number of approach lanes.

More information about the operation of junction modelling within the dynamic traffic assignment model OtMadam can be found in § 3.3. Do not get confused by the display in the Junction Editor. It will always show a schematic view showing one lane on the link.

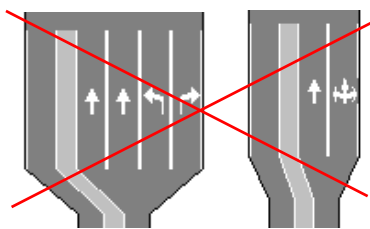
#### *Length of the approach lanes*

For static traffic assignment modelling, the approach lanes theoretically have an infinite length (or a so-called vertical queue). This means that blocking back will not be taken into account at static assignment modelling.

For dynamic traffic assignment modelling, either a default length (for the whole network) or the length of the underlying link is determinative. More information about the operation of junction modelling within the dynamic traffic assignment model OtMadam can be found in § 3.3.

#### *Crossing movements*

Lane configuration in which turning movements cross each other are not allowed (see Figure 9). Bus and special lanes @@are excluded from this rule.



*Figure 9 - Crossing movements are not allowed.*

#### *One-way links*

Lanes to arms without an exit lane, i.e. one-way links, or to non-existing arms are not allowed. When a one-way link enters or exits a junction, the Junction Editor automatically disables the input fields for respectively the approach lanes or the number of exit lanes.

One remark about this has to be made. The definition of a junction is only valid for one mode.

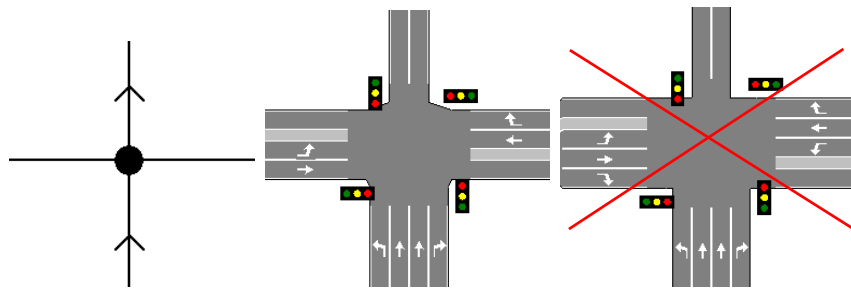


Figure 10 - Example of junction with one-way entry and exit arms.

The righthand junction in Figure 10 generates an error during the initialisation (or check) of the junction modelling. It will report that there are so-called ‘redundant’ lanes, i.e. the left and right turning lanes to the south (see § 2.1.7).

*Omitting lanes*

It is not allowed to omit lanes for specific turning movements unless a banned turn has been defined (see § 2.2.4).

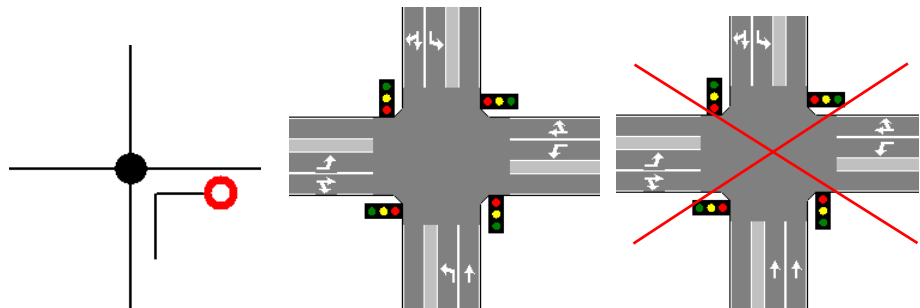


Figure 11 - Example of junction with banned turn.

The righthand junction in Figure 11 generates an error during the initialisation (or check) of the junction modelling. It will report that there is a missing left-turn lane from the arm from the south (see § 2.1.7).

*Entry lanes for roundabouts*

Approach lanes for roundabouts should be defined as approach lanes for a normal junction, i.e. with left, straight and right turn lanes.

This is important for the utilisation of the lanes by the different movements. Left turning traffic primarily uses the left most entry lane whereas right turning traffic primarily uses the right most entry lane.

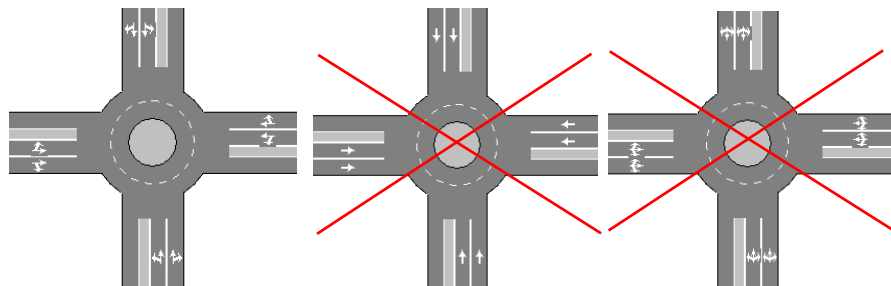


Figure 12 - Examples of entry lanes for roundabouts.

For a one lane approach, a ↕ lane configuration is usual, for a two lane approach a ↕↕ configuration is common.

*What is straight and what is turning?*

In some situations it is difficult to determine which movement is left, straight or right due to small or large angles between the arms of the junction. OmniTRANS allows you to define the junction as you want it, i.e. as it is in real world. Figure 13 shows two examples of the same junction which are all coded differently and correctly.

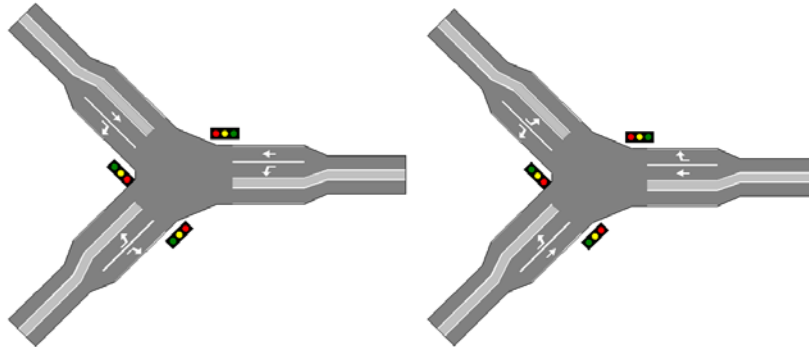


Figure 13 - Examples of different left, straight and right turning lane configurations.

*Special lanes*

OmniTRANS can model ‘special lanes’ such as a bus priority lane or a slip lane (unsignalised right turn). Special lanes are defined with a separate character in the lane configuration. Currently, the junction modelling module can handle the following types of lanes:

Key	Symbol	Description
B		Bus lane. Only relevant for signalised junctions and roundabouts. When the Junction Type is not manual, an extra green phase with minimum green time is introduced for the bus, when one or more bus lanes are present. A bus lane may be used at any position in the lane configuration, e.g. BLSR, LSBR or LSRB.
o	o	Slip lane. Only relevant for signalised junctions and roundabouts. A slip lane can be used to reckon with the fact that right turning traffic can use a separate lane which is not included in the signal scheme controlling the junction. The right turning traffic is assumed to have to give way. Slip lanes are ignored in the lane groups definition for signals. A slip lane may be used at any position in the lane configuration, however, it is advisable to code it at the right side, e.g. LSSo or LSRo. For left hand drive rule projects, the slip lane can be used to model unsignalised left turning traffic.
1-9	1..9	Reserved for future <sup>3</sup> use.

Table 2 - Description of special lanes.

<sup>3</sup> The next special lane type implement in OmniTRANS will enable you to control flared approaches.

### Slow Traffic

The slow traffic option is used to indicate whether there is crossing pedestrian and/or bicycle traffic on the arm. Although the option is available for all Junction Types, it is only relevant for signalised junctions and roundabouts. When there is at least one arm with crossing slow traffic, an extra phase is introduced during the calculation of automated signal types. This phase has a minimum green time (see § 2.2.2).

### Central Reservation

The width of the central reservation (or mid verge), i.e. the space between the entry and exit lanes (default: 0.0 meters). The width of the central reservation is used in various calculations concerning blockade probabilities.

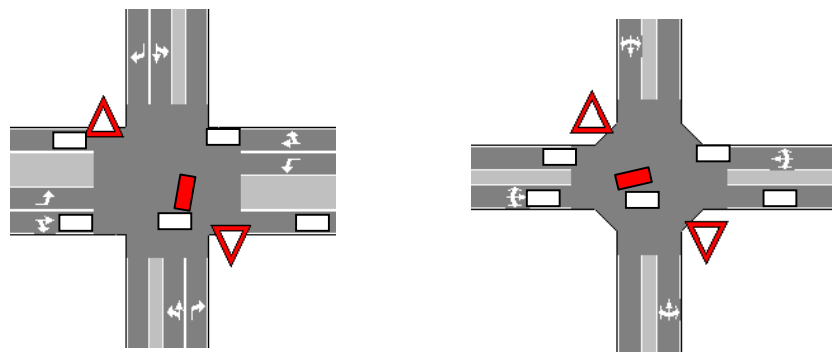


Figure 14 - Effect of the width of the central reservation.

Figure 14 shows two examples of the importance of a reasonable width for the central reservation. In the left hand junction, the central reservation is used to cross the junction in two stages. In the right hand junction, the central reservation is used as a buffer area for waiting left turn vehicles, so that the straight moving vehicles can proceed without extra delay.

### Exit Lanes

The number of exit lanes from the junction on to the arm currently being defined (default: 1 lane). You can define a maximum of 6 exit lanes. The number of exit lanes is important for the calculation of the capacity for a specific turning movement, especially for unsignalised junctions.

It is not possible to define the number of exit lanes on arms which are one-way approaching links (because there is no street at all leaving the junction). It is possible and allowed to define the number of exit lanes on arms with only banned-turns to it (see Figure 15). Although this might not always be a realistic situation, the junction modelling does not generate an error on with this coding.

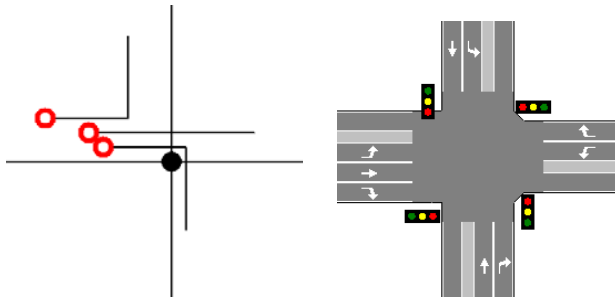


Figure 15 - Banned turns and exit lanes.

### Enabled

The junction modelling of OmniTRANS can accommodate up to four arm junctions. The reason for this is that on a five arm junction, it is not possible to say to which arm the right turning, straight or left turning movements are going to, since there can be multiple destination options.

If the junction has more than four arms you either re-code the network so that all arms can be modelled explicitly (e.g. to decompose a complex roundabout into a series of give-way junctions), or you must nominate which of the arms (> four) are the least important and can be ignored. These arms would have the 'Enabled' option set to 'No'. The junction should further be coded as if the disabled arms were not there. For arms that are disabled, no turning delays are calculated.

It is not allowed to disable arms on junctions with less than five arms, or to enable less than four arms.

This can only be done for the static assignment modelling. In the case of dynamic modelling, all junction must have a most four arms. Disabling arms is not allowed.

### 2.1.6 Signal Settings

When the Junction Type is a signalised junction or a signalised roundabout, it is also possible to access and edit the signal settings of the junction. This can be done on the Signal Settings tab of the Junction Editor. When there is no Signal Settings tab in the Junction Editor, the project has to be upgraded first (see § 2.2.6).

The Signal Settings tab can be used both for viewing output as well as for generating signal settings input. This section describes the input functionalities.

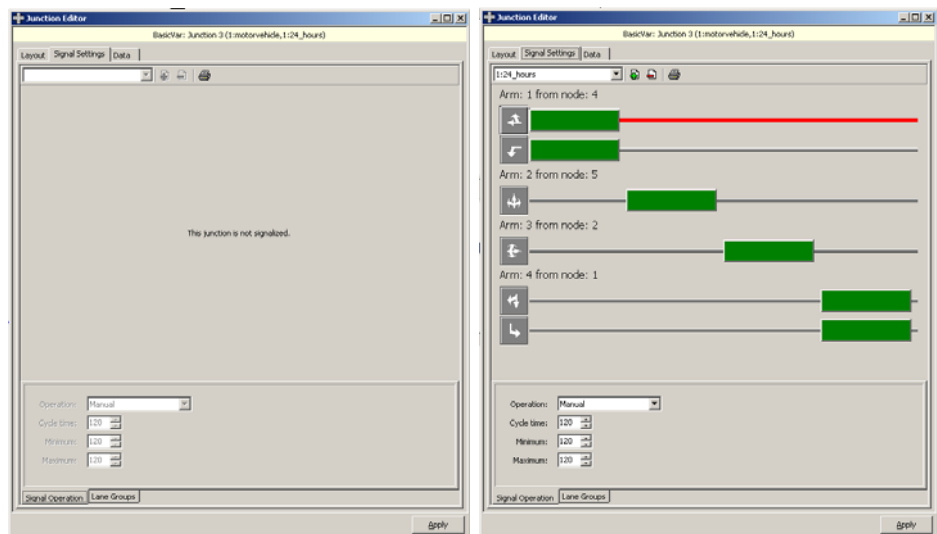


Figure 16 - Signal settings tab for both unsignalled and signalled junctions.

## Signalling Schemes

OmniTRANS works with signalling schemes. A signalling scheme is a complete set of signal settings associated with a particular time interval. A junction can have multiple signalling schemes attached to it, e.g. for the morning peak, for the off-peak and for the evening peak. The time interval can also vary between schemes, e.g. 9:00-9:15 am. When you add a new signalling scheme, you have to specify the start and end time, thus defining the time interval for which the scheme applies. You can choose the start and end time from a drop-down list. The available options correspond with the time dimension items that have been defined in the Project Setup of your project (see Figure 17), noting that the specifications of the time periods for static assignment modelling and dynamic modelling is different. See below

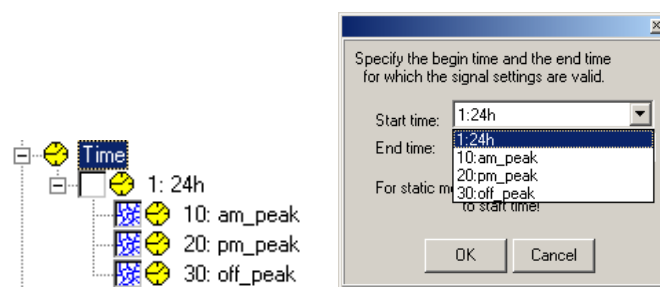


Figure 17 - Time dimension items in Project Setup and start/end time options for signalling schemes.

### Schemes for static models

If you define a signalling scheme for use in a static assignment, the start and end time **must** be the same, since it is an average state for the given time period. So, if you want to run an assignment for time 10 (e.g. morning peak), you must define a signalling scheme with start and end time equal to 10.

### Schemes for dynamic models

For dynamic models, you can define as many different schemes as necessary, even with a different duration for each scheme. You can define a scheme for 7:00-8:00, 8:00-8:30, 8:30-8:45, etc.<sup>4</sup> The dynamic assignment model automatically switches between the schemes during the assignment process (see § 3.3).

Signalling schemes can be added, deleted or modified. There is no limit to the number of schemes that can be attached to a junction. When you create a new signalling scheme, OmniTRANS makes a copy of the currently visible signalling scheme. If the created scheme is the first one for this junction, the scheme gets default values for the cycle time and green times. A scheme can also be based on the cycle time and green times that are stored in an existing 'load'. Simply choose the related 'pmturi' from the list of available schemes. You can modify an existing scheme by simply choosing it from the list of available signalling schemes.

The remaining options can be defined for each signal scheme that has been made for the junction. That includes the signal type, so you can e.g. define an 'automated' signal in the peak period and a 'manual' signal in the off-peak.

### Signal Type

OmniTRANS offers the following signal control types:

- Manual (fixed-time, pretimed);
- Automated (optimised);
- Actuated (cycle time actuated);
- Actuated (vehicle-actuated).

#### *Manual*

This means that you have coded a fixed-time (or pretimed) signalling scheme. The cycle time and green times will never change for the period for which the scheme is defined.

#### *Automated*

This means that you do not code any cycle time or green times. The junction modelling module calculates an optimal (minimum average control delay) cycle time and green times for the given junction layout and turning flows. This calculation can be repeated each iteration (static assignment) or time step (dynamic assignment).

#### *Actuated (cycle time actuated)*

This means that you have coded average, minimum and maximum cycle time. The junction modelling module changes the cycle time given the coded minimum and maximum boundary, to minimise the average control delay. This calculation can be repeated each iteration (static iteration) or time step (dynamic assignment).

#### *Actuated (vehicle-actuated)*

---

<sup>4</sup> For the time being, OtMadam only looks at the start time of a signal scheme. An example: If you want to define a scheme for the period 7:00-9:00 and a specific scheme for 8:00-8:30, you'll have to define 3 schemes, namely 7:00-8:00, 8:00-8:30 and 8:30-9:00 to get the required result.

This means that you have coded average, minimum and maximum cycle time and green times. The junction modelling module changes the cycle time and green times given the coded minimum and maximum boundaries, to minimise the average control delay. This calculation can be repeated each iteration (static iteration) or time step (dynamic assignment).

The availability of the following options is dependent upon the choice of signal control type. In case of an automated signal control, all options are disabled. In case of a fixed-time signal control, the minimum and maximum cycle time and green times are disabled. In the case case of actuated signal cycle time actuated, the minimum and maximum cycle time are enabled. Finally, in case of a vehicle-actuated signal control, all options are enabled.

### **Cycle Time**

The length of one signal cycle (seconds). Time required for one sequence of signal displays (sum of phase green and intergreen times). For a given movement, cycle time is the sum of the duration of red, yellow and green signal displays, or the sum of the effective green and red times. Typical values are 60,90 and 120.

### **Minimum and Maximum Cycle Time**

The minimum and maximum allowed cycle time (seconds), used for actuated signal control types. The minimum value must be a positive value, the maximum value may not be lower than the average cycle time. The maximum value overrules the global maximum cycle time.

### **Lane Groups**

The green times in a signal scheme are assigned to lane groups. A lane group is a set of lanes with one or more shared lanes (e.g. lane 1: left-turn and through, lane 2: through) or a set of exclusive turn lanes (e.g. a single right-turn lane).

OmniTRANS automatically generates the lane groups on the basis of the definition of the approach lanes in the junction layout (see § 2.1.5). This is based on the concept that each turning movement generates a separate lane group, unless there are shared lanes.

In the latter case, the turning movements cannot have different signal displays, and thus are they combined into one lane group. Each lane group is displayed as a separate graphical line (left-side) and a separate tab (lower right-side). The first lane groups starts at the right-side of the road of the first arm. The numbering continues clockwise. Figure 18 shows an example of a junction and the lane groups that are generated for this junction layout.

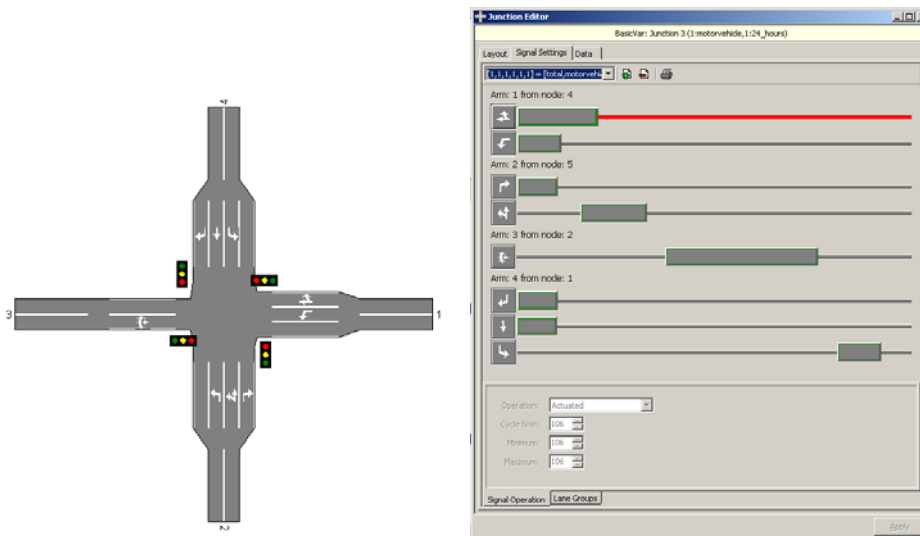


Figure 18 - Examples of different types of lane groups generated from the junction layout.

The attributes of each lane group can be accessed separately by selecting one of the tabs or by clicking on the associated line in the graphical area of the tab.

### Offset

The offset is used to indicate the starting time of the effective green for this lane group, related to the starting time of the signal cycle (seconds). An offset of 20 seconds, means that the green time for this lane group starts at 20 seconds from the beginning of the signal cycle. Lane groups with the same offset and the same green time, technically form a signal phase, although OmniTRANS is not restricted to phasing.

### Green Time

The effective green time of the lane group (seconds). This includes the corrections for the start loss and end gain times (see § 2.2.2). The rest of the cycle time, the display for the associated lane group is red. Lane groups with the same offset and the same green time, technically form a signal phase, although OmniTRANS is not restricted to phasing. See Figure 19 for an example with four signal ‘phases’.

It is easy to define a signal scheme in which one of the lane groups in a phase starts earlier and/or continues longer. It is also possible to define additional green time that is not adjacent, i.e. the lane group is part of multiple green phases. The latter can be defined by the ‘Addition Green’ button, which is explained later in this section.

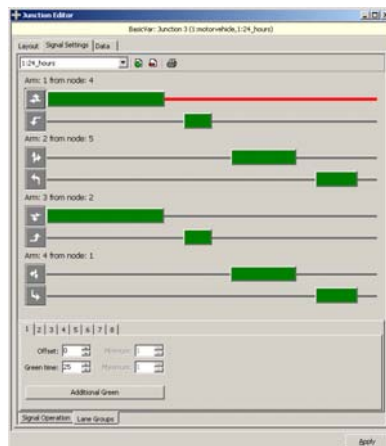


Figure 19 - Signal scheme with four phases.

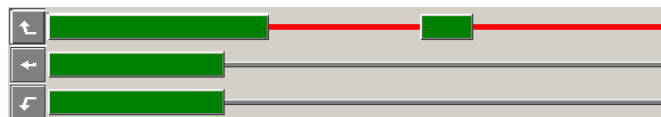


Figure 20 - Example of extended green and additional green time.

#### Handling Bus Lanes and Pedestrians

Bus lanes are not translated to separate lane groups. They are only used for the calculation of an optimised (automated) signal. If you want to code a specific green time for buses, you simply code a period of all-red, i.e. no green for any lane group. The same applies to green time for crossing pedestrian or bicycle traffic.

#### Minimum and Maximum Green Time

The minimum and maximum allowed effective green time (seconds). Used for actuated signal control types only. The minimum value must be a positive value, the maximum value may not be lower the average effective green time. The minimum value overrules the global minimum green time for a turning movement or lane group.


#### Additional Green

A lane group can be assigned two green times that are not adjacent, i.e. have a different offset (typically have green in two different phases). This can be done by pressing the 'Additional Green' button. For the addition green time, a new block of green time is drawn and a new tab is created. The tab gets the number of the lane group plus an 'A'. The additional part can be dealt with in a normal fashion.

#### 2.1.7 Check Button

If you press the Check button, either on the Layout tab or the Signal Settings tab, you'll get a report about possible errors in the definition of the layout or the signal schemes, e.g. that the approach lane definition causes crossing flows or that a lane is missing or that the minimum cycle time is higher than the maximum. The messages are similar to the ones that are generated during the initialisation stage of the assignment process and will be explained in more detail in the next Chapter (see § 3.3.4). There can however be differences in the messages in the Junction

Editor and those generated during the initialisation stage of the assignment process. This can for instance be caused by a different choice of network and/or specific assignment properties like 'skipLinks'.

In some cases, the report mentions node numbers. These can be displayed in the Junction Editor window by enabling the second  button on the Layout tab.

### 2.1.8 Examples

This section describes some examples which deal with 'Frequently Asked Questions'.

#### Large (Gyratory) Roundabouts

The predefined roundabout type of OmniTRANS is intended for modelling small (or mini-) roundabouts with a maximum of four arms and two circulating lanes on the roundabout.

Behaviour on larger roundabouts is different and thus should be dealt with in a different manner. Large (gyratory) roundabouts should be modelled by coding the roundabout in the network and each entry/exit as a separate T-junction. Each T-junction can then be coded as a priority or signalised junction (see Figure 21). In the example, traffic on the roundabout has priority over traffic entering the roundabout. It is best to code the exit's and entry's as one link and thus as one combined T-junction instead of as separate T-junctions or as one four arm junction.

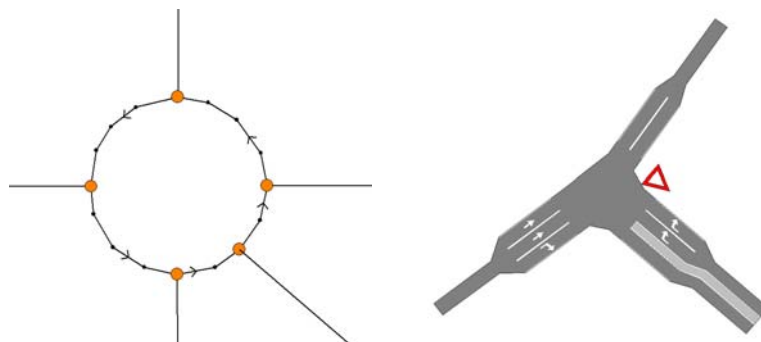


Figure 21 – Large roundabout coded in the network.

#### Junctions with more than Four Arms

The predefined Junction Types in OmniTRANS can accommodate up to four arms per junction. If a junction has more than four arms there are two ways to deal with this situation, namely:

- Disable one or more arms in the Junction Editor;
- Code the junction as two or more separate junctions in the network.

The first option is only used when a fifth or sixth arm plays no significant importance for the modelling of the flows on the junction. You can then simply disable the arms with the 'Enabled' option in the Junction Editor (see § 2.1.5). The

junction should further be coded as if the disabled arms were not there. No turning delays are calculated for turn coming from or going to the disabled arms. You can however code delays on turns manually (see § 2.2.4).

The second provides a better solution. Now one junction is simple split up into two or more separate junctions. Figure 21 for example already showed a solution for a five arm roundabout. Figure 22 shows a standard example for a five arm signalised junction.

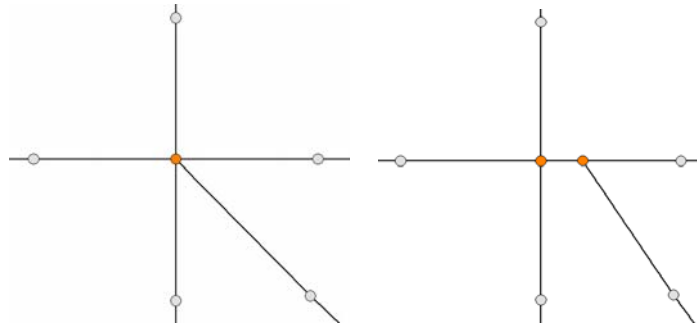


Figure 22 – Five arm signalised junction wrongly and correctly coded in the network.

### Junction By-passes

A junction ‘by-pass’ is where a lane is provided for a given turning movement which ‘by-passes’ the prime junction control. OmniTRANS provides two ways to deal with junction by-passes:

- Use the special ‘slip lane’ letter/number in the Junction Editor;
- Code the bypass in the network.

The first option is only meant for signalised junctions and/or roundabouts where a short ‘slip’ lane is available for turning traffic which can ‘ignore’ the traffic signal and just has to give way. The ‘slip’ lane has a two vehicle storage space, which are addition to the normal lane(seconds) available for right turning traffic. The slip lane is typically used for dynamic traffic assignment modelling, since coding a very short by-pass in the network would disturb OtMadam’s modelling theory and the handling of the lane utilisation in the approach lanes.

The second option is meant for all other situations, typically for larger by-passes. The bypass is then simply coded in the network as a separate (one-way) link. You can control the length, capacity and type of control at the end of the by-pass yourself. Figure 23 shows an example of a bypass on a small roundabout.

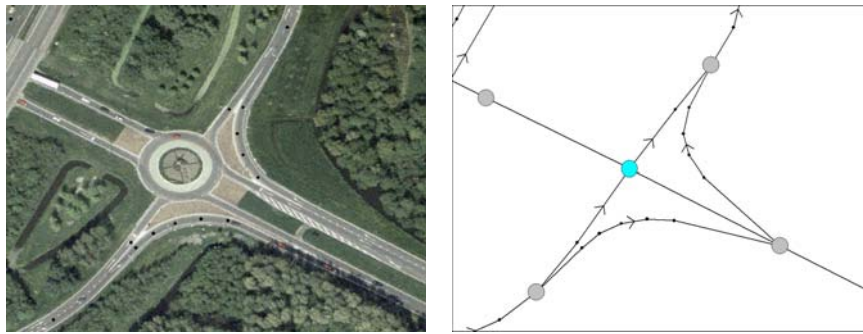


Figure 23 – Example of coding of bypasses at a roundabout.

### Multiple Modes on a Junction

Junction modelling in OmniTRANS is designed to determine the turning delays for motorised traffic. Although it can account for the presence of pedestrians and/or bicycle crossings and bus priority lanes, it does not determine delays or any other performance measures for these modes of transport. The networks in OmniTRANS can however be multi-modal, which means that not all links have to be available for motorised traffic. How should you deal with that while coding junctions?

The Junction Editor has no knowledge about the modes of transport that may or may not be using the links to and from a junction. It therefore always shows all available links, also when these are not accessible for motorised traffic, e.g. being a bicycle track. If the links entering or exiting a junction are not available for motorised traffic (for the mode you are going to use the network from during the assignment), you should just ignore these entry or exit lanes in the coding of the junction. Figure 24 shows an example of a four arm junction, where one of the arms is a bicycle track. The junction is simply coded as a three arm junction.

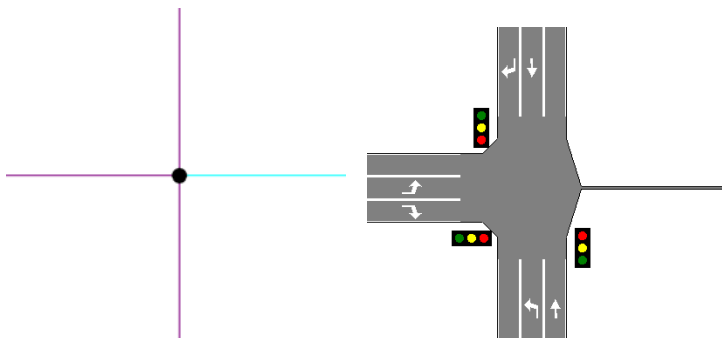


Figure 24 – Junction with bicycle track (from the East).

### Junctions and Connector Links

Centroid connectors should NOT be connected to a (modelled) junction as these will destroy the junction delay calculations which are based on (true) link flows.

### Pedestrian Crossings

Stand-alone pedestrian or bicycle crossings cannot be modelled using one of the Junction Types. However, it is possible to manually define turn delays, e.g. an average delay of a couple of seconds to simulate pedestrian and/or bicycle

crossings, also on non-junction nodes. This will be explained in more detail in § 2.2.4.

## 2.2 Advanced Input

In order to simplify the coding of the junctions in the Junction Editor, OmniTRANS provides a set of default values both for the general behaviour on junctions as well as for the modelling of the traffic flows on the predefined Junction Types. These default values are stored in a separate Junction Definition File and can be accessed and changed by the user. This section explains the contents of the Junction Definition File and how it can be changed. It also discusses the use of the Network Editor besides or instead of the Junction Editor to code banned turns or to overrule calculated saturation flows or delays. The section concludes with an explanation of the database tables and fields which are used to store the junction data and how you can upgrade old projects to the new database structure.

### 2.2.1 Junction Modelling Defaults

A design consideration for junction modelling in OmniTRANS was that the time and effort for coding the junctions in the network should be minimised, subject to a high quality of the produced output and without compromising modelling flexibility. This resulted in the use of default values, both generally as well as for a series of predefined Junction Types.

The default values can be controlled by editing an ASCII-file in which the default values are stored. Changing options in an ASCII-file is not ideal, however the format is rather straightforward and will be explained in this section. Keep in mind that editing such a file brings about a certain amount of responsibility for the correctness of the syntax of the file.

The junction modelling defaults are stored in a file called 'junctions.def' which can be found in the main installation folder of OmniTRANS (e.g. 'c:\Program Files\Omnitrans International\OmniTRANS 4.2\'). The settings in this file apply to all OmniTRANS projects that are modelled with that particular version of OmniTRANS. The default values for junction modelling in the file always starts with a section header called '[Junction Modelling]'.

Alternatively, the contents<sup>5</sup> of the 'junctions.def' file can be copied into the 'project.gcp' file, which can be found in the main folder of each OmniTRANS project. In this way, the defaults can be altered and stored for that particular project separately. It is however recommended to leave the junction modelling defaults in the 'junctions.def' file. Also, keep in mind that after installing a new OmniTRANS version, which has been installed in a separate folder, the altered 'junction.def' must be re-processed into it to keep previous settings.

Another possibility is to place the 'junctions.def' file in the project directory itself. In this case, first the global will be read and then the local file. This means that only the updates have to be defined in the local file.

---

<sup>5</sup> Including the '[Junction Modelling]' section header.

The junction modelling defaults consist of two different types of data:

- Project data;
- Junction Type data.

### 2.2.2 Project Defaults

This comprises a series of default values for all junctions in the project. The format of these variables in the file is like:

```
MinGreenTime = 6
MaxCycleTime = 120
```

Although the default syntax of the variable names uses both lower and upper case, the notation is not case sensitive.

#### PeriodDuration

The duration of the analysis period for junction modelling (default: 1 hour). This value is used for the calculation of the average control delay for each turn or lane (see § 5.2).

The following three default values are used to determine the average lost time between two successive phases of a traffic signal. The average lost time is determined by the inter green time minus the end gain time plus the start loss time. The average lost time is calculated for junctions with an automated and actuated signal operation type (see § 2.1.6) in order to determine the optimal cycle time and green times.

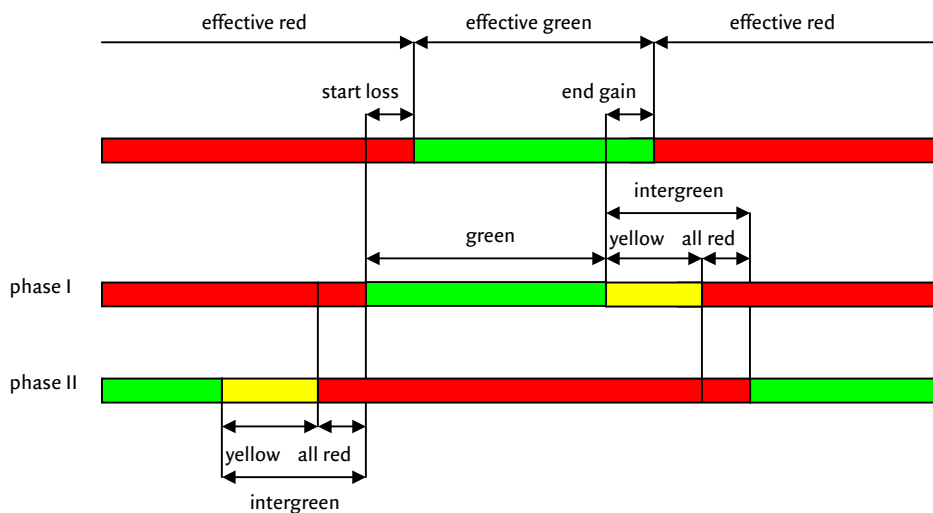


Figure 25 - Definitions of signal settings defaults.

### StartLossTime

The start loss time. The first few seconds of the green time, which are not effectively used due to the need to react to the initiation of the green phase and to accelerate (default: 2 seconds). See Figure 25.

### EndGainTime

The end gain time. The first few seconds of the yellow time which are effectively used by vehicles to cross the junction (default: 3 seconds). See Figure 25.

### InterGreenTime

The intergreen time. This is the yellow plus all-red time that occurs between phases of a traffic signal (default: 4 seconds). See Figure 25.

The following three default values are used for the calculation of the optimal cycle time and green times for a junction with an automated or actuated signal operation type (see § 2.1.6).

### PracSatDegree

The practical degree of saturation. This is used as a target (maximum acceptable) degree of saturation for signal timing calculations (default: 85%), which is used for the determination of the optimum cycle time.

### MinGreenTime

The minimum allowed effective green time for a phase or turn (default: 6 seconds). This value is also used for the length of a possible extra phase for crossing pedestrians, bicycles and/orbusses.

### MaxCycleTime

The maximum allowed cycle time (default: 120 seconds). Calculated optimal cycle times might in some situations be rather high. Although this might be optimal for the flow of traffic, relatively long red times do not ‘feel’ optimal for travellers. Long red times might engender red light negation and consequently cause precarious situations. Use of a maximum cycle time gives more realistic results.

The following two default values are mainly important for unsignalised junctions.

### MinCapacity

The minimum allowed calculated capacity for a turn (default: 100 pcu/hour). This value is mainly important for unsignalised junctions, where traffic has to give way for a large stream of traffic. Although the normal priority rules and the capacity formulas might imply a zero capacity, in reality a certain amount of vehicles is often “allowed” to merge or cross a major stream.

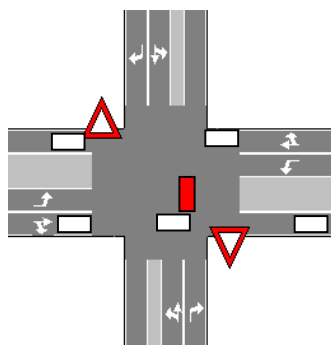


Figure 26 - Queue space on the central reservation for two-stage crossing.

**VehQSpace**

The length of road a vehicle in a queue uses (default: 5 meter). This value is used to determine whether a vehicle can use the central reservation in order to perform a two-stage crossing of opposing traffic flows for unsignalised junctions. See Figure 26.

The following default value is used for all Junction Types, although it is mainly applicable for use in a static assignment.

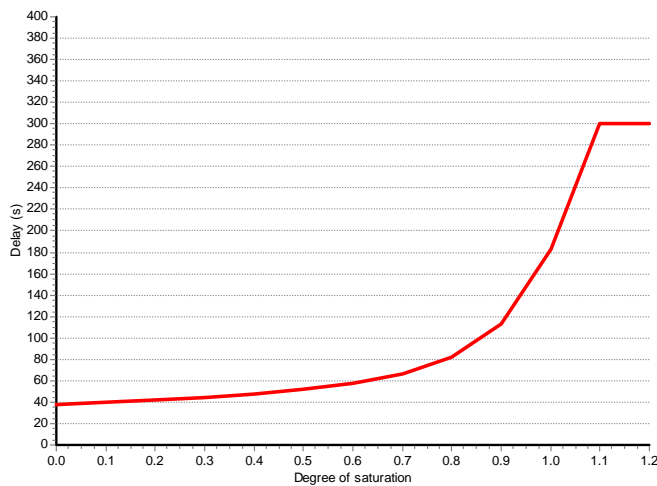


Figure 27- Delay versus degree of saturation (maximum delay is 300 seconds).

**MaxDelay**

The maximum allowed delay for a turn (default: 300 seconds). In static assignment modelling, the calculated delays may be very high, especially in the first iteration of an equilibrium (/volume averaging/capacity restraint) assignment, where all traffic for a relation is assigned to one shortest route. This can create unrealistic oversaturated situations on junctions and thus extremely high delays.

Experience has revealed that using a maximum delay to top off the delay curve (see Figure 27) speeds up the convergence process, or in case of volume averaging or capacity-restraint assignments improves the results.

**2.2.3 Junction Type Defaults**

As has been explained earlier, the use of default values was mainly introduced to speed up the coding of the junctions in the network. Besides some general default values, OmniTRANS also uses default or predefined Junction Types. By choosing a specific Junction Type in the Junction Editor window (see § 2.1.2), the junction automatically gets a whole series of default values, like the lane width, saturation flows for turning movements or priority rules for conflicting flows. By means of editing the Junction Type data these defaults can be changed.

The quantity and the assigned numbers for the predefined Junction Types are fixed. At the moment OmniTRANS knows the following types:

1. Equal (or uncontrolled) junction;
2. Priority junction;
3. Signalised junction;
4. Roundabout;
5. Signalised roundabout;
6. All-way-stop junction.

The format of the predefined Junction Type values looks like:

```
TypeDef4_Nr = 4
TypeDef4_Name=Roundabout
TypeDef4_Roundabout=1
TypeDef4_Signal=0
TypeDef4_LaneWidth=3.5
TypeDef4_RACentralDiameter=18
```

Each value will be explained briefly in the remaining part of this paragraph.

### Nr

The number of the Junction Type. At the moment, this is a unique numbering that should not be changed by the user.

### Name

The name of the Junction Type. Note that this name is not (yet) used in the Junction Editor window.

The following two values are very important for the modelling process, because they determine to which of the four main junction modelling categories the Junction Type belongs.

		signalised	
		no	yes
roundabout	no	unsignalised junction	signalised junction
	yes	unsignalised roundabout	signalised roundabout

*Table 3 – Base junction modelling types.*

### Roundabout

Whether the junction is a roundabout or not. Options:

- 0 No roundabout;
- 1 Roundabout.

### Signal

Whether the junction is signalised or not. Options:

- 0 No signal (unsignalised);
- 1 Signal (signalised).

The following two values only concern unsignalised junctions.

**SignType**

The type of signs at the junction in case of an unsignalised junction. Options:

- 0 None;
- 1 Yield signs;
- 2 Stop signs.



Figure 28 – Priority sign types (none, yield, stop).

The location and number of the signs can be controlled by way of the Junction Editor (see § 2.1.2).

**PriorityType**

Whether the direction of the major road for a priority controlled junction is straight or turning. Options:

- 0 None;
- 1 Straight;
- 2 Turning.

This value is merely used to distinguish between the earlier mentioned predefined Junction Types 2 and 7.

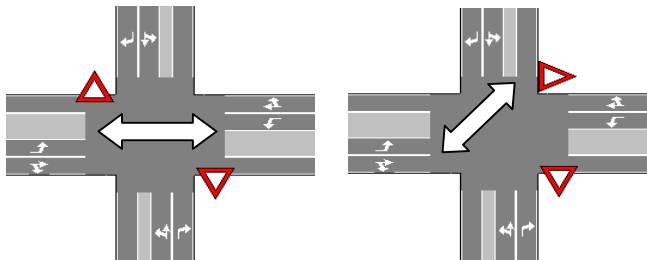
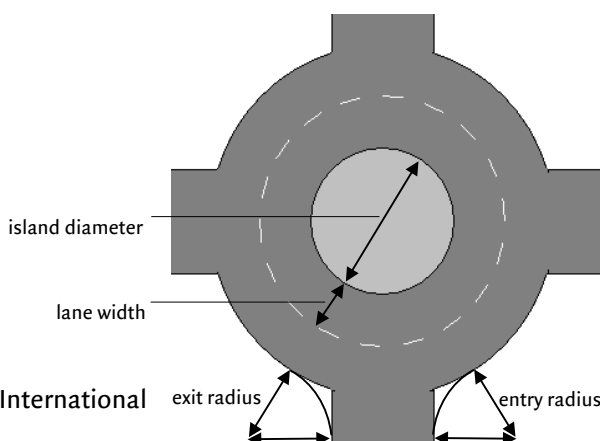


Figure 29 - Straight or turning priority.

The following seven values only concern roundabouts. These values are used to determine the available distance between the entry and exit lanes of the roundabout. The distance is in turn used to determine the magnitude of the effect of so-called semi-conflicting flows. A semi-conflict arises when traffic on the roundabout does not give an indication of direction when it leaves the roundabout. Traffic that wants to enter the roundabout and has to give way to traffic on the roundabout (unnecessarily) waits for the semi-conflicting flow. This increases the average delay.



*Figure 30 - Definition of roundabout geometry defaults.*

**RACentralDiameter**

The central island diameter of a one lane roundabout (default: 15 meters).

**RAEntryRadius**

The one lane roundabout entry radius (default: 12 meters).

**RAExitRadius**

The one lane roundabout exit radius (default: 15 meters).

**RACentralDiameter2lanes**

The central island diameter of a two lane roundabout (default: 40 meters).

**RAEntryRadius2lanes**

The two lane roundabout entry radius (default: 12 meters).

**RAExitRadius2lanes**

The one lane roundabout exit radius (default: 15 meters).

**RALaneWidth**

The width of each circulating lane on a roundabout (default: 5.5 meters).

The following three values apply for all Junction Types.

**LaneWidth**

The average width of an approach or exit lane (default: 4.25 meters).

**SatFlow**

The basic saturation flows (pcu's/hour). OmniTRANS uses basic saturation flows for each movement per Junction Type. Saturation flows can be differentiated by:

- Left-turning movement;
- Straight movement;
- Right-turning movement;
- Exit movement.

In contrast with other packages in OmniTRANS you do not have to give the saturation flow for each turn or lane as input for each junction. OmniTRANS uses the default values and decreases the saturation flow automatically accounting for effects like the number of lanes, shared lanes, give way, blockade probabilities, signal settings, etcetera.

The saturation flows are stored like:

```
TypeDef4_SatFlow(0)=1700,1800,1800,1800
```

The values are ordered according to the above mentioned enumeration.

It is also possible to overrule the default saturation flows for each turn on each junction separately. This can be done by defining saturation flow values for each turn on the junction. If a turn has a value for the saturation flow, the default values are not used (see § 2.2.4).

**Conflict**

Conflict matrices for both right- and left hand side rules. For unsignalised junctions and roundabouts, the conflict matrix is used to determine to which turning movements a specific turning movement has to give way. For signalised junctions and roundabouts, the conflict matrix is used to determine the turning movements which can have green at the same time and which not.

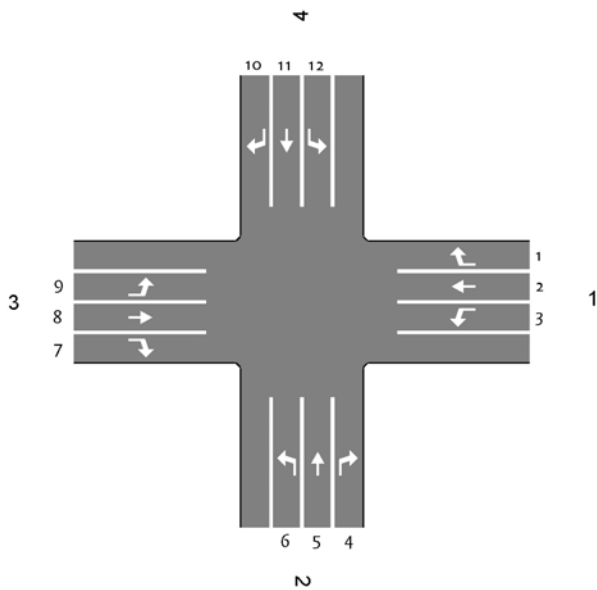


Figure 31 - Numbering of turning movements.

The conflict matrix is a 12x12 matrix where the rows and columns represent the different turning movements on a standard four-arm junction (see Figure 31). Turning movements 1,4,7 and 10 are right turns, turning movements 2,5,8 and 11 are straight turns whereas turning movements 3,6,9 and 12 are left turns.

The following figure shows a few examples of conflict matrices for the following Junction Types:

1. Uncontrolled junction;
2. Two-way priority junction (main road from arm 1 to 3);
3. Roundabout;
4. Signalised junction (no conflicts allowed).

(1)	(2)	(3)	(4)
123456789012	123456789012	123456789012	123456789012
1 ----12--1---	1 ----1---1222	1 -----222	1 ----1---1---
2 ----11--1---	2 ----11--1111	2 -----222111	2 ----11--1111
3 ----11--1---	3 ----11----11	3 ---222111111	3 ---1111---11
4 -----12--1	4 -----1	4 222-----	4 -----1----1
5 -----11--1	5 -----1	5 111-----222	5 111---11---1
6 -----11--1	6 -----	6 111---222111	6 -11---1111--

7 --1-----12	7 --1222----1-	7 ---222-----	7 --1-----1-
8 --1-----11	8 --1111----11	8 222111-----	8 --1111----11
9 --1-----11	9 ----11----11	9 111111---222	9 11--11----11
0 -12--1-----	0 ----1-----	0 -----222---	0 -1--1-----
1 -11--1-----	1 ----1-----	1 ---222111---	1 -11--1111---
2 -11--1-----	2 -----	2 222111111---	2 -1111--11---

1 = conflict  
2 = semi-conflict

*Figure 32 - Examples of conflict matrices.*

In the first conflict matrix, for an uncontrolled (or equal) junction, one can read, by looking at the first column in the matrix, that turning movement 1 does not have to give way to any other turning movement. Turning movement 2 on the other hand, has to give way to all traffic from the right, i.e. turning movements 10,11 and 12. Turning movement 6 can have a semi-conflict with turning movement 1, e.g. when there is a shared lane for movements 1 and 2/3.

For signalised junctions, the conflict matrices are used to determine which turning movements can be combined in one signal group or phase and which not. Conflicting turning movements cannot be combined. In the fourth conflict matrix, one can read, by looking at the first column in the matrix, that turning movement 1 cannot be combined with turning movements 5 and 9.

A conflict matrix is stored as one line of data. The first part of the conflict matrix for the predefined roundabout type looks like:

```
TypeDef4_ConflictR=00000000022200000022211100022211111122200...
```

The first 12 characters represent the first row of the matrix. A separate conflict matrix is stored for both right- and left-hand side rules.

### BlockProb

Blockade probability matrices for both right- and left hand side rules. The calculated capacity for each turning movement on unsignalised junctions (not roundabouts) is multiplied by the probabilities that the movement is not blocked by stationary waiting vehicles from other movements. The blockade probability matrix determines which probabilities should be applied to which turning movements.

The following figure shows examples of a blockade probability matrix for the following Junction Types:

1. Uncontrolled junction;
2. Two-way priority junction (main road from arm 1 to 3).

<p>(1)</p> <p>123456789012</p> <p>1 -----</p> <p>2 -----11</p> <p>3 -----12-11</p> <p>4 -----</p> <p>5 -11-----</p> <p>6 -11-----12</p> <p>7 -----</p>	<p>(2)</p> <p>123456789012</p> <p>1 -----</p> <p>2 -----</p> <p>3 -----2---</p> <p>4 -----</p> <p>5 --1-----1---</p> <p>6 --1-----1112</p> <p>7 -----</p>
--	---

8	----11-----	8	-----	
9	-12-11-----	9	--1-----	
0	-----	0	-----	
1	-----11---	1	--1-----1---	1 = always
2	----12-11---	2	--1112--1---	2 = hooked left-turning movements

Figure 33 - Examples of probability matrices.

The calculated capacity of turning movement 6 on a uncontrolled junction is multiplied by the no-blockade probabilities of turning movements 2, 3 and 11. It is also multiplied by the probability of turning movement 12 if opposite pairs of left-turn movements ‘hook’, i.e. interfere. OmniTRANS automatically determines whether the situation requires a not hooked or hooked solution.

The following two sets of values represent a series of modelling parameters which are used in the delay and queue equations for each Junction Type. The delay and queue equations within OmniTRANS have been generalised, such that one can set the parameters according to the national or local preferences, such as the HCM2000.

**DelayPar**

Five parameters for the delay formula. See § 5.2 for the use of the parameters in the generalised delay equation.

Parameter	OmniTRANS default	HCM2000
<i>A</i>	0	0
<i>B</i>	0	0
<i>K</i>	1	0.5
<i>M</i>	8	8
<i>N</i>	0	0

Table 4 - Delay parameters.

The parameters a and b are only used for signalised junctions and roundabouts (see § 5.2). The syntax for the DelayPar values is:

```
TypeDef2_DelayPar(0)=0,0,1,8,0
```

**QueuePar**

Five parameter for the queue formula. See § 5.3 for the use of the parameters in the generalised queue equation.

Parameter	OmniTRANS default	HCM2000
<i>a</i>	0	0
<i>b</i>	0	0
<i>k</i>	1	0.5
<i>m</i>	8	8
<i>n</i>	0	0

Table 5 - Queue parameters.

The parameters a and b are only used for signalised junctions and roundabouts (see § 5.3). The syntax for the QueuePar values is:


```
TypeDef2_QueuePar(0)=0,0,1,8,0
```

#### 2.2.4 Network Editor

The Network Editor can of course be used to code the network nodes and links that form the basis for the junctions. It can also be used to code turn prohibitions or to overrule the calculated turn saturation flows and/or delays. This will be explained in the first part of this section. The Network Editor can also be used as an alternative to the Junction Editor for editing some of the node and link data. This can for instance be used to define multiple junctions at the same time. This will be explained in the second part of this section.

##### Turn Data

The turn object in OmniTRANS is based on three nodes, a from-node, a through-node and a to-node. Turns can be displayed by means of the normal OmniTRANS settings in the Graphics toolbar (see the OmniTRANS manual).

Turn objects are either created by the user or by the Junction Editor. In the first case, a turn is added by the user e.g. in order to define a turning prohibition or a specific saturation flow for a turning movement. In the latter case, a turn object is created for each possible turning movement on a junction once it is assigned a particular Junction Type. It is necessary to create these objects, since OmniTRANS needs them to store both junction input and output data. Turns that are created by the Junction Editor are not visible in the Network window unless a particular option is enabled. Their visibility is controlled by the  button from the Turn Settings in the Graphics toolbar. Figure 34 shows an example of the visualisation of turn objects that have been generated by the Junction Editor.

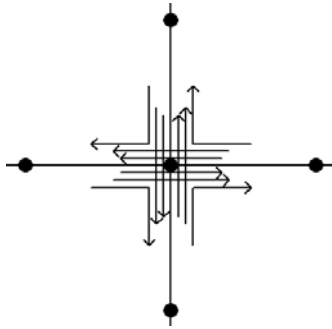


Figure 34 – Turn objects that have been generated by the Junction Editor.

If you add a new turn object on a node that has been defined as a junction, you will receive the message ‘This turn might affect a junction. Make sure that the junction data is correct’. The turn is added, but it is up to the user to ensure that the junction definition is still correct. This will depend upon the (default) properties of the turn just added. The most important issue is whether the new turn is a banned turn or a turn with some ‘overruling’ values for the calculated saturation flows and/or delays of the turn.

If the ‘impedance’ value of a turn is equal to -1, it is a banned turn. This means that there must be no approach lanes for this particular turning movement in the junction definition (see § 2.1.5). The matter is slightly complicated by the fact that the ‘impedance’ value can be differentiated by mode and time, which could mean that a turn is banned for car traffic but not for bicycle traffic or a turn is only banned in the morning peak and not in the evening peak. A banned turn is always visualised with a stop sign at the end of the turn, whereas a ‘normal’ turn is visualised with an arrow (see Figure 35).

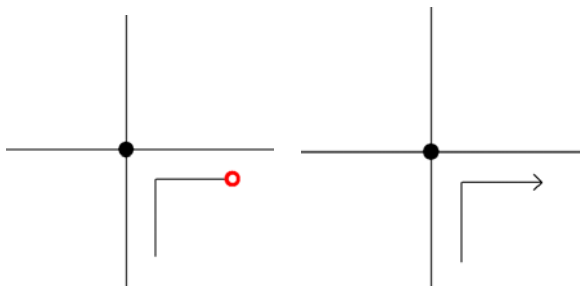



Figure 35 – Banned turn and ‘normal’ turn.

In order to access the properties of a turn, simply select it and open the Attribute Editor. You can change one or more impedance and saturation flow values.

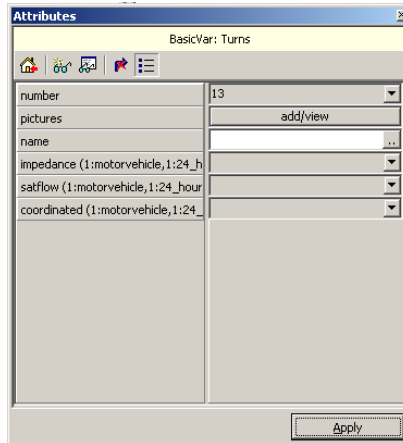
#### Impedance

The impedance field can be used for two purposes. If the impedance value is set to -1, the turn is banned for that particular mode and time. If the impedance value is set to zero or higher, it represents the average control delay on the turn for that particular mode and time. The unit of time is seconds. The impedance value overrules the delay value that is calculated by the junction modelling module.

A value of zero is different then no value at all. Zero means no delay on this turn, and no value means that junction modelling will calculate the delay.

So, if you want to remove a particular impedance, but want to preserve the turn object, you can use the  button in the Attribute Editor.

You can also code a turn on locations where there is no actual junction, but e.g. a pedestrian crossing and define an average control delay for the car traffic on that particular point (see Figure 36).



*Figure 36 - Example of a turn on a non-junction node.*

### Saturation Flow

The saturation flow field is used to overrule the default saturation flow values that are used by the junction modelling module. Normally, these values are based on the Junction Type and the type of movement (see § 2.2.3). The saturation flow (or ‘satflow’) represents pcu’s/hour for one lane for that particular type of movement.

### Node and Link Data

The normal way to define junctions in OmniTRANS is by using the Junction Editor. Alternatively, the Network Editor can be used to access and change the junction layout attributes. These attributes are stored on nodes and links and can be accessed by way of the Attribute Editor. By default, the Attribute Editor does not visualise the junction layout attributes. To activate the visualisation of these attributes you will have to change a setting in the ‘project.gcp’ file which can be found in the main folder of your project. Set ‘EditJunctions=1’ under the ‘[Defaults]’ section header. For example:

```
[Defaults]
EditJunctions=1
```

Be sure that you change the option when OmniTRANS is not running, or at least not with the concerning project opened.

Note that, due to historical reasons, the names of the fields in the Junction Editor and in the Attribute Editor (and the database) might be different.

### Node Data

On a node, the following fields are available for junction definition:

- JunctionType, the Junction Type;
- RALanes, the number of circulating lanes on a roundabout;
- CoordFactor, the coordination/calibration factor.

These fields have been explained in § 2.1.3 and § 2.1.4.

The Attribute Editor can be used to code a particular Junction Type for multiple nodes at the same time. This is not possible with the Junction Editor. In the Attribute Editor this can simply be done by selecting multiple nodes and changing the 'JunctionType' in the Attribute Editor (and of course press the Apply button). Mind that you still have to code the appropriate approach lanes on each junction. By default, each arm will receive one lane which is available for all turning movements.

*Link Data*

On a link, the following fields are available for junction definition:

- LanesMask, the definition of the approach lanes;
- WidthCR, the width of the central reservation;
- ExitLanes, the number of exit lanes;
- SlowTraffic, the indicator for a pedestrian and/or bicycle crossing;
- Sign, type of sign at the end of the link;
- Enabled, whether the arm is enabled or not.

These fields have been explained in § 2.1.6, although the 'Sign' field was not mentioned as such. The Junction Editor lets you choose a certain type of priority as a property of the junction. The type of sign (yield or stop) is determined by the Junction Type. This information is stored on links. A 0 value means that there is no sign, a value of 1 stands for a yield sign, a value of 2 for a stop sign.

It is important to understand that all junction related attributes are stored on the link direction that is going to the junction (in-link). This means that both the approach lanes as well as the exit lanes are stored on the same in-link.

**2.2.5 Database**

This paragraph gives a brief overview of the database tables and fields that are used to store the junction input information. For a full explanation of the database, see TN003.

The relational database of an OmniTRANS project consists of three levels, respectively project, variant and sub-variant level. General junction input data is stored on the variant level. Table 6 gives an overview of the database tables and fields that are used for the junction input data.

Table	Fields
Node1data1	<i>Nodenr</i> , junctiontype, ralanes, coordfactor
Node1_1data1	<i>Nodenr</i> , starttime, endtime, signaltype, cyletime, maxcyletime, mincyletime
Link1_1data1	<i>Linknr</i> , <i>direction</i> , lanesmask, widthcr, exitlanes, slowtraffic,

	sign, enabled
Turn1_3data1	Turnnr, starttime, endtime, greenoffset, greentime, mingreentime, maxgreentime
Turn3data1	Turnnr, mode, time, impedance, satflow

*Table 6 – Database tables and fields for junction input data.*

The database tables and fields that are used to store the junction modelling output will be discussed in § 4.5.

### 2.2.6 Upgrading Old Projects

Between OmniTRANS version 4.1 and version 4.2, an extension of the database structure for junction modelling has been introduced. New tables were added and existing tables were extended. Projects that are created with a project template from version 4.2, will automatically have the new database structure. Projects that have been created in older versions will not automatically get the new database structure once they are opened in version 4.2. This is done for reasons of backwards-compatibility. You can use the older projects in OmniTRANS 4.2, but you will not have the ability to define signal schemes and have a limited number of performance measures for a junction. If you want to have these options in an older project, you will have to ‘switch the template’, i.e. upgrade your project.

The easiest way to recognise whether you have an old or new database structure is by looking at the Junction Editor. If the Junction Editor does not have a ‘Signal Settings’ tab, then the structure is old.

If you have your project open in OmniTRANS 4.2, you can upgrade your project by simply choosing the ‘Switch template’ option from the Project – Templates menu. Choose either a static or dynamic template from the set of installed templates. Your existing project setup, type, zonal data fields, etcetera will not change. OmniTRANS only adds the new database tables and fields. You can check whether the operation was completed successfully by looking for the Signal Settings tab in the Junction Editor.

## 3 Modelling

This Chapter gives detailed information on how to activate and control junction modelling within both the static and dynamic traffic assignment modelling classes of OmniTRANS, respectively OtTraffic and OtMadam. These classes are activated by way of jobs.

The first section explains the general junction modelling properties that can be used to activate and control the junction modelling for both static and dynamic traffic assignment modelling. The subsequent two sections describe how the junction modelling works within the process of respectively a static and a dynamic traffic assignment and which specific properties are available. The Chapter ends with an explanation of the error and warning messages that can be produced by the junction modelling module using OtTraffic, OtMadam or the Check button in the Junction Editor.

### 3.1 General Junction Modelling Properties

Junction modelling in OmniTRANS can be used within the static (OtTraffic) or the dynamic (OtMadam) traffic assignment model. Both OtTraffic and OtMadam are modelling classes which have the following format:

```
assign = OtTraffic.new
...
assign.execute
```

Between the new and the execute methods, a series of properties must and/or can be defined to control the assignment. These can be junction modelling properties. For both classes, there are three general class properties which are relevant to control junction modelling:

- junctions;
- junctionParameters;
- junctionVersion.

#### junctions

This property specifies whether or not junction modelling is activated. When you want to activate junction modelling, set this property to true. For example:

```
assign.junctions = true
```

The default value is false.

#### junctionParameters

This property specifies a set of conversion factors to convert the assigned trip volumes to a reference of passenger car units (pcu's) per one hour. These factors are necessary as junction modelling outputs (e.g. delays, capacities) are calculated on the basis of pcu's per hour.

The default formulation is:

```
assign.junctionParameters = [1.0,1.0]
```

Where the first number specifies the peak hour conversion factor and the second number the pcu conversion factor. So if the OD-matrix that is used in the assignment contains motor vehicles for a two hour peak period, the conversion factors could be:

```
assign.junctionParameters = [0.5,1.2]
```

It must be clear that the conversion factors are not applied on the OD-matrix itself. The factors are applied on the turn flows which are generated by the assignment model and are input to junction modelling (for each junction separately). This is purely an internal matter, which means that the flows that are stored on each link and turn, as well as the calculated capacities are stored with the same units as the OD-matrix. The conversion factors are merely used to calculate an average delay (and queue) for each turn.

### junctionVersion

This property specifies the version of the junction modelling calculations to be used if junction modelling is invoked. During the development of the junction modelling module the algorithm has been improved and will be improved in future version. As the changes can have impact on the results of a model you can preserve backwards compatibility, if required, by using this parameter. An example:

```
assign.junctionVersion = 4
```

The default is always the latest version. You can see which junction modelling version an assignment model is using by looking at the output messages at the Job Engine Output window. The junction modelling version is reported under the Input (Reading network...) section. For example:



Figure 37 - Visualise which junction modelling version is used.

A brief report on the differences between the different versions can be found in the OmniTRANS manual. A more in depth report can be found in Appendix A of this document.

## 3.2 Junction Modelling within a Static Assignment

The junction modelling feature can be combined with any static assignment method, although it is usually used in combination with a capacity restraint method.

This section first of all explains what is the place of junction modelling within the assignment model in general. Subsequently, it describes how the junction modelling feature operates in combination with various assignment methods and discusses topics like “How the junction delays are reckoned with in the generalised cost formulation?”, “How/if the junction delays are stored in output skim matrices?”, “How multiple purposes, modes or user-classes are dealt with?”, et cetera.

### 3.2.1 Traffic Assignment and Junction Modelling

Independent of the assignment method, the process of using junction modelling within the assignment process is similar. The basic steps followed during an assignment process are:

1. Read all networkdata (link, node, turn) from the database;
2. Build the network (including junctions) in the computer's memory;
3. Check the network (including junction definitions) and report errors and warnings;
4. Calculate cost on links and turns based on a free-flow costs (zero loads);
5. Determine the shortest paths for all OD-pairs;
6. Assign the loads to the shortest paths for all OD-pairs;
7. Recalculate the costs on links and turns based on the newly calculated loads.

What follows after step 7 is dependent upon the assignment method that is chosen.

In step 3, the assignment model performs a check on the network data, including the junction definitions. If something is wrong with the coding of a junction, it will produce one or more warnings and the junction will be excluded from further calculations. Junction warnings are not critical, which means that the assignment process will continue despite the fact that one or more junctions are coded inaccurately. The junction will simply be skipped during the process of calculating the turn cost, thus leaving the cost equal to zero. It is highly recommended that all junction warning messages should be solved. These warnings will be explained in § 3-4.

After step 3, the junction layout and the signal settings of the junction are fixed. The only input that changes each time junction modelling is used, are the turning flows of the junction. In step 4, the flows are still zero, which means that the resulting turning delays (or costs)<sup>6</sup> are also equal to zero. In step 7, loads have been generated for each turning movement, resulting in the calculations of turning costs at each junction.

---

<sup>6</sup> OmniTRANS stores the resulting turning delays in the 'cost' field. Therefore, we speak of turning cost instead of turning delays.

In general, the assignment model generates turning flows and the junction modelling module uses these to generate turning cost. Besides the turning costs, junction modelling also generates various other junction and turning values, but these will be discussed in Chapter 4.

### 3.2.2 All-or-Nothing Assignment

An all-or-nothing assignment works according to the seven steps described in the previous paragraph. A minim job example:

```
assign = OtTraffic.new
assign.load = [1,1,1,1,1,1]
assign.junctions = true
assign.execute
```

Although costs (and other values) are calculated for each turning movement, the shortest paths have been calculated on the basis of a free-flow situation, thus with turning costs equal to zero. If turning costs have been coded manually (see § 2.2.4), these are taken into account in the shortest path generation.

Another exception to the rule that no turning costs are included in the shortest path generation, is the use of the `initialLoad` property. In that case, no free-flow but a loaded situation is used to calculate the first set of link and turn costs (step 4). An example:

```
assign = OtTraffic.new
assign.load = [1,1,1,1,2,1]
assign.junctions = true
assign.initialLoad = [1,1,1,1,1,1]
assign.execute
```

### 3.2.3 Incremental Assignment

An incremental assignment is an iterative assignment. In each iteration a portion of the trips for each OD-pair is assigned to the shortest route. The first iteration follows the earlier described seven steps, with the exception that not all but only a portion of the total number of trips is assigned to the network (e.g. 20%). After step 7, the process loops back to step 5 and a second portion of the traffic is assigned to the network in step 6 (e.g. again 20%). The junction turning costs are thus recalculated at the end of each iteration and are used to determine the new shortest paths in the next iteration. An example:

```
assign = OtTraffic.new
assign.load = [1,1,1,1,1,1]
assign.assignMethod = INCREMENTAL
assign.iterations = 5
assign.junctions = true
assign.execute
```

In the example, in each iteration 20% of the trips between each OD-pair is assigned to the network. It is also possible to differentiate the proportions for each iteration. See the OmniTRANS manual for more information.

### 3.2.4 Volume-Averaging Assignment

The volume-averaging assignment method works according to the same principle as the incremental method. The main difference between the two methods lies in the fact that the volume-averaging assignment assigns 100% of the trips in each iteration and averages the loads at the end of each iteration. The link and turn costs are recalculated after the averaging process. A typical feature of a volume-averaging assignment is that each iteration has the same weighting in the total process (1/number of iterations). For more information see the OmniTRANS manual.

An example:

```
assign = OtTraffic.new
assign.load = [1,1,1,1,1,1]
assign.assignMethod = VOLUMEAVERAGING
assign.iterations = 10
assign.epsilon = 0.001
assign.bprPerType = [[[1..10],[0.5,4.0]]]
assign.junctions = true
assign.execute
```

The epsilon that is reported during the assignment process, and which is used as a stop criteria, includes turning costs for junctions.

### 3.2.5 Equilibrium Assignment

The equilibrium assignment method, which estimates a user equilibrium according to the Frank-Wolfe algorithm, is very similar to the volume-averaging method. The main difference is the weighting of the results of each iteration. The weight is now calculated by the algorithm so that results that are closer to the final user equilibrium are more important. For more information see the OmniTRANS manual. For now, it is sufficient to know that both link as well as turn costs are used in the function to determine the weight for each iteration.

### 3.2.6 Generalised Cost Formulation

Shortest paths are searched on the basis of so-called generalised cost. Generalised cost can be a composed of a combination of distance, time and/or various cost elements. In OtTraffic this is controlled by the routeFactors property.

```
assign.routeFactors = [0,1,0,0]
```

The property contains a set of four weighting factors, respectively for distance, travel time, additional fixed link cost and fixed cost from a previous assignment. For an assignment with junction modelling it is important to know that the calculated turn delays (or cost) are multiplied by the weighting factor for time. The other three factors are not relevant for turn costs.

### 3.2.7 Skim Generation

OtTraffic can produce three different skim matrices simultaneously. These matrices contain respectively the travel distances, times and generalised cost for each OD-pair. The travel time skim matrix incorporates the calculated junction turning delays (or costs). Depending upon the settings of the routeFactors property, the turning delays (or costs) are also incorporated in the generalised cost skim. See the OmniTRANS manual for more information.

### 3.2.8 Multi-User-Class Assignment

If you want to assign multiple purposes, modes and/or user-classes, mostly in a volume averaging or equilibrium assignment, you can use one of the following methods in OmniTRANS:

- Reduce the capacity with an existing load before performing a capacity-restraint assignment;
- Simply assign multiple matrices in one go. All matrices are however assigned to the same routes;
- Perform a multi-user-class assignment, where the choice behaviour and thus the shortest routes may be different for each purpose, mode and/or user.

Each method has different consequences for the quality of the junction modelling and will therefore be explained briefly in the remainder of this section.

#### Reduce capacity

This is mostly done for a combination of freight and car traffic. Freight traffic is then assigned with the all-or-nothing method. Subsequently, car traffic is assigned with a capacity-restraint method, where the capacity of the car network is reduced with the freight load prior to the actual assignment. The latter can be done with the OtTraffic changeCapacity property. With the changeCapacityFactors property, you can also control a multiplication factor (e.g. to convert trucks to pcu's) for the load that is used to reduce the capacity. For example:

```
assign.changeCapacity = [[1,50,10,1,1,1]]  
assign.changeCapacityFactors = [1.5]
```

It is also possible to use multiple loads and factors, e.g. when you differentiate heavy and light freight. For more information, see the OmniTRANS manual.

A major drawback of this method is that only link capacities are reduced. Turn capacities are not influenced by the changeCapacity property, neither do the turn loads incorporate the freight traffic. This means that the turn loads used in the junction modelling are underestimated as do the resulting turn costs (delays). It is advised not to use this method in combination with junction modelling.

#### Assign multiple matrices

Another approach would be to assign multiple OD-matrices in one assignment run. The loads on each link will then be summed and normal capacity restraint effects on links and junction modelling calculations can be done. Given that the OD-

matrices are given in the same vehicle units, e.g. pcu's, this gives normal junction modelling behaviour. The resulting loads are stored in separate pmturi's. A drawback of this method is that the trips for each OD-matrix are assigned to the same set of routes, i.e. if you assign car and freight traffic, you can not differentiate route choice behaviour between these two modes.

This method can be implemented by way of the load and/or odMatrix properties of OtTraffic. You are allowed to define multiple items for the Purpose as well as the User dimension, or both. Some examples:

```
assign.load = [1,1,1,[21,22,22],1,1]
```

```
assign.load = [[1,2],1,1,[21,22,22],1,1]
```

```
assign.load = [1,[10,20],1,1,1,1]
```

The last example is not allowed. If you want to assign multiple OD-matrices for modes, you will have to temporarily move them to a Purpose or User dimension item.

### Multi-user-class assignment

The third, and definitely best but also most complex way to model multiple purposes, modes and/or user-classes is by way of a true multi-user-class assignment job. The OtTraffic class offers several options to let you build a job that performs a multi-user-class capacity restraint assignment with junction modelling. The OmniTRANS manual offers an extensive example of such a job. It lies outside the scope of this document to go into to much detail about this.

In short, you are performing all-or-nothing assignments for each user-class in a loop, combining the loads of the classes and previous iterations. The calculated link and turn costs are used for the shortest path calculations of the next iteration. Because this is partly done outside OtTraffic, you have full control over the properties of each user-class. Different route choice behaviour and network availability is possible for each class. It is up to you to define whether different purposes, modes and/or user-classes should be used.

A limitation of this method is that at the moment it cannot be combined with an equilibrium assignment method.

## 3.3 Junction Modelling within a Dynamic Assignment

Besides the static assignment methods, OmniTRANS also offers a macroscopic dynamic assignment method by way of the OtMadam class. OtMadam can use the same junction definitions as the static methods. with dynamic modelling, some specific rules apply to the junction modelling, such as the recalculation of the junction turn costs (or delays) in time.

This section explains how the junction modelling functions within the dynamic assignment process. OtMadam offers various methods of dynamics concerning the recalculation of the shortest paths and junction turning costs. These methods are then described from a junction modelling point of view.

### 3.3.1 Dynamic Traffic Assignment and Junction Modelling

It is important to understand some of the basics principles underlying the dynamic assignment methodology. OtMadam lets traffic flow through the network by splitting the links into small segments and recalculating the current stage for each segment every one to five seconds. The stage is represented by a speed-flow-density relationship which in its turn is influenced by the stage on adjacent up- and downstream segments. If the density on a downstream segment increases, the speed on a segment is dropping, as is the (out)flow. If the density on a segment exceeds the maximum value (by default 180 vehicles per kilometre) the ‘queue’ blocks back to the upstream segment. Special equations are used for links with merges, weaving and lane-drops and of course junctions.

Route choice in OtMadam is based on turn fractions for each node. These turn fractions can be made prior to a run of OtMadam, e.g. by using a static traffic assignment result (load), or can be recalculated by OtMadam itself dynamically. It is also possible to differentiate the fractions by mode (or purpose, and/or user-class) For more information, see the OmniTRANS manual.

With dynamic modelling, there are a couple of points that are explicitly important for junction modelling.

#### Calculations by lane

On the segment approaching a junction, special rules apply. Next to the calculated cost for each turning movement, the junction modelling module also produces average delays and capacities for each approach lane separately. Each lane is dealt with as a separate (parallel) segment. The lane delay is translated to an average speed on the segment while the lane capacity is used one-on-one. The junction modelling also delivers the distribution of the different turning movements over the lanes. Based on all this information and the speed-flow-density relationship, OtMadam can determine the stage on the segment in front of the junction. If one of the approach lanes reaches the maximum allowed density, the queue will block back on the segment upstream similar to a situation on normal links and segments.

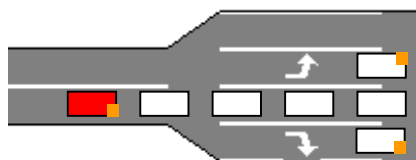


Figure 38 - Approach lane blocking back from junction.

### Lane length

In contrast to static assignment, the length of the approach lane(s) is important. The length of the approach lane(s) is determined either by a default value defined for OtMadam or by the length of the link which approaches the junction. The default length of the approach lanes can be controlled using the modelParameters property of OtMadam. This property uses a list of general modelling parameters. The 9<sup>th</sup> parameter in the array represents the default length for approach lanes (or junction segments).

```
assign.modelParameters = [[20,35,13,0.8,2,1,180,0.3,0.1],1]]
```

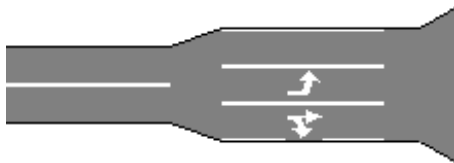
In this example, the default length of the approach lanes is set to 0.1 distance units (usually kilometres). This value is applied for the whole network, except at locations where the approaching link is shorter than the supplied default value.

It is (not yet) possible to differentiate the length of lanes at one approach, i.e. the right, straight and left turning lanes all have the same length.

### Link lanes and approach lanes

There has to be a certain level of consistency between the number of lanes on the link and the number of approach lanes. E.g. it is not allowed to have more lanes on the link than there are approach lanes on the arm. OtMadam will generate a warning message about this prior to the assignment.

Suppose a link of one lane with a length of 500 meters, connected to a junction. The two approach lanes (left and combined straight-right) have a length of 50 meters. The first 450 meters are then dealt with as a normal stretch of road (one lane), while the last 50 meters are modelled as junction approach lanes. OtMadam deals with the effects of the extra lane(seconds).



*Figure 39 - One lane link and two approach lanes.*

When the length of the link is smaller than the default value for the length of the approach lanes, the number of lanes on the link becomes irrelevant.

Do not be confused by the display in the Junction Editor. It will always show a schematic view showing only one lane on a link. The situation is modelled correctly, this is only a visualisation matter.

### Junctions at motorways

OtMadam has special ways to deal with on- and off-ramps of motorways as well as weaving segments. This means that these situations should not be coded as junctions in the network. OtMadam will automatically detect these situations. For more information, see the OmniTRANS manual.

### Recalculation of junction costs and route choice

OtMadam allows you to control whether you want to recalculate the junction costs (lane delays and capacities, lane load distribution) and/or route choice within an assignment run. Various combinations, as well as the possibility for a dynamic multi-user-class assignment will be discussed in the subsequent paragraphs. This document only explains the importance of each method for junction modelling, so the explanations will be very brief. For a more detailed description, see the OmniTRANS manual.

#### 3.3.2 Static Route Choice and Junction Modelling

This is the most basic way to use OtMadam. The dynamic model uses an existing static assignment to determine the turning fractions and the lane data for the junctions. The fractions and junction data are calculated once, at the start of the assignment and stay constant throughout the whole run. The stage (speed-flow-density relationship) on each approach lane is of course recalculated every time step (typically one to five seconds). A job could look like:

```
assign = OtMadam.new
assign.staticLoad = [1,1,1,1,1,1,1]
assign.odMatrix = [1,1,1,1]
assign.duration = 8*15
assign.departureFractions = [0.05,0.2,0.25,0.2,0.15,0.1,0.1,0.05]
assign.load = [1,1,1000,1,1,1]
assign.junctions = true
assign.junctionParameters = [0.5,1.2]
assign.execute
```

The junctionParameters are applied to the turn volumes (loads) that are coming from the staticLoad, thus converting two-hour vehicles to pcu's per hour.

In this example, we explicitly do not use the urbanLinkTypes property. Only use this property in very special situations. For more information see the OmniTRANS manual.

A general drawback of a method using static junction modelling is that the calculated junction delays and capacities are insensitive to the actual number of traffic crossing the junction. This is especially poor for automated signalised junctions, since the cycle time and green times can then be based on situations which do not apply for the whole modelling period, thus causing unnecessary queues.

### 3.3.3 Static Route Choice and Dynamic Junction Modelling

The previous example can simply be extended to a model in which the junction data is recalculated each time step (typically one to five seconds). You can simply add the `dynamicJunctionUpdate` property to the list.

```
assign.dynamicJunctionUpdate = true
```

The model still uses a static assignment result to determine the route fractions and the initial junction data, but now recalculates the junction data every time step. Instead of using the turning flows from the static load table, the junction modelling module is now fed with turning flows from the OtMadam calculations. Since these flows are always in pcu's/hour the junctionParameters are not used after the first recalculation (i.e. set to 1.0 and 1.0).

A major drawback of this method is that it doesn't recalculate any routes. If a junction causes a lot of delay, it will not cause people to change another route to there destination.

### 3.3.4 Dynamic Route Choice and Junction Modelling

The next step is to let OtMadam recalculate the shortest routes every given time step. This time step is different from the time step (typically one to five seconds) that is used to recalculate the situation on each link and junction. The recalculation step for route choice is e.g. 15 minutes.

This means that the turning fractions are re-determined every given time step. To do this, OtMadam performs an all-or-nothing assignment on the basis of the costs at that particular moment in time (including the newly calculated turn delays for a junction).

### 3.3.5 Dynamic Multi-User-Class Assignment

OtMadam also allows you to perform a dynamic assignment for multiple purposes, modes and/or user-classes. Since this is done internally within the OtMadam class, you don't have to build your own multi-user-class assignment as is the case for static assignment modelling. Therefore, it can easily be combined with dynamic route choice and dynamic junction update. Junction modelling simple uses the sum of different modes, purposes and/or user-classes. See the OmniTRANS manual for more information.

## 3.4 Error and Warning Messages

This section describes the error and/or warning messages that are generated by the junction modelling module, either in a job by running OtTraffic or OtMadam or by pressing the Check button in the Junction Editor.

All junction modelling error messages are officially warning, which means that they are not fatal for the assignment modelling process. If one or more warnings are generated for a particular junction, it is simply excluded from further calculations. No output will be generated for the particular junction.

Most warning messages are generated during the initialisation stage of OtTraffic or OtMadam, when the data is read from the database and checked for consistency. A similar check is done for a separate junction when the Check button in the Junction Editor is pressed. There can be differences in the messages reported by the Check button and the ones posted by OtTraffic or OtMadam. A difference can be caused by the choice of network or by the exclusion of certain links from the assignment by using the skipLink method. Pay close attention to these types of junction. Sometimes, you'll have to code the junction 'wrong' in the Junction Editor if you want it to pass the check in OtTraffic or OtMadam.

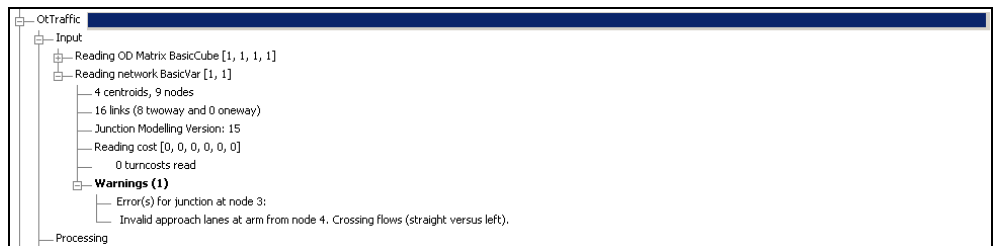


Figure 40 - Example of a junction modelling warning message.

A warning message about a junction always starts with the text “Error(second) for junction at node...”, followed by one or more specific warning messages. The remainder of this paragraph describes these specific warning messages in alphabetic order.

### Cycle time is zero or negative

The cycle time for a signalised junction is not allowed to be zero or negative. Although the Junction Editor does not allow you to enter a zero or negative value for the cycle time, it might be possible that the database contains a negative value, e.g. due to import or manual editing.

### Error in one or more approach lanes

When the junction modelling module cannot interpret the node and link data as a proper junction definition and does not know which other error to post, it will generate this error message. The message occurs rarely, and is mostly generated when the project database is corrupted or sometimes on junctions where there are no conflicts (T-junctions with one-way links (or e.g. bicycle tracks), and/or banned turns). Perform a ‘Check Database...’ from the main Project menu or simply set the junction to ‘<undefined>’.

### Fatal error during the calculation of the junction

If something goes wrong in the algorithm which performs the junction modelling, it will report this message. The calculations for this particular junction are aborted and no turn delays (or costs) will be reckoned with. Perform a ‘Check Database...’ from the main Project menu or simply set the junction to ‘<undefined>’ and re-define it again. If this doesn’t help, contact your distributor or Omnitrans International for further assistance.

### **Fatal error during the initialisation of the junction**

If something goes wrong in the algorithm which performs the checks for the junction definitions, it will report this message. The initialisation for this particular junction is aborted and no turn delays (or costs) will be reckoned with. Perform a 'Check Database...' from the main Project menu or simply set the junction to '<undefined>' and re-define it again. If this doesn't help, contact your distributor or Omnitrans International for further assistance.

### **Green time of turn (FROMNODE,TONODE) is zero or negative**

The green time of a turn is not allowed to be zero or negative. Although the Junction Editor does not allow you to enter a zero or negative value for the green time, it might be possible that the database contains a negative value, e.g. due to import or manual editing.

### **Invalid approach lanes at arm from node FROMNODE.**

OmniTRANS does not allow you to code approach lanes so that one or more turning flows (from one arm) have to cross each other to pass the junction. The second part of the message indicates which flows are crossing each other:

- Crossing flows (right versus left/straight);
- Crossing flows (straight versus left).

### **Invalid calibration factor. Value must be positive.**

The calibration factor of a junction is not allowed to be negative. Although the Junction Editor does not allow you to enter a negative value, it might be possible that the database contains a negative value, e.g. due to import or manual editing.

### **Invalid number of arms. A junction must have at least three arms.**

OmniTRANS does not support junction definitions for nodes with less than three arms. Set the Junction Type to <undefined> or ignore the warning message.

### **Invalid number of enabled arms. A junction with more than four arms must have four enabled arms.**

A junction with five or more arms must have at least four enabled arms, otherwise this message will be posted. The enabled option is only meant to reduce the number of arms to exactly four arms. See § 2.1.5 and § 2.1.8 for more information.

### **Invalid number of lanes on the roundabout. Value must be 1 or 2.**

OmniTRANS only support roundabouts with one or two circulating lanes. If you want to model a roundabout with three or more circulating lane, you'll have to code the roundabout in the network as separate T-junctions. See § 2.1.5 and § 2.1.8 for more information.

### **Invalid number of stop signs.**

The number of stop or yield signs is not correct. Dependent upon the number of arms on the junction and the type of the junction, the second part of the message reports the number of expected signs.

For a yield sign controlled junction, for respectively three and four arm junctions, the second part of the message can be:

- Number must be 1;
- Number must be 2.

For a stop sign controlled junction, for respectively three of four arm junctions, the second part of the message can be:

- Number must be 1 or 3;
- Number must be 2 or 4.

**Invalid signal type. Type number is unknown.**

This message is posted when the database contains a signal type number that is not within the range of allowed numbers (0,1,2). Although the Junction Editor does not allow you to enter an invalid signal type, it might be possible that the database contains an incorrect value, e.g. due to import or manual editing.

**Invalid turn. From node of turn (FROMNODE,TONODE) is not found in arms.**

The junction modelling module has received a turn object which is not connected to the junction, or the from node of the turn is not used for this junction. This error only occurs when there is corrupt or invalid data in your project database. Perform a 'Check Database...' from the main Project menu or simply set the junction to '<undefined>' and re-define it again.

**Invalid turn. Load of turn (FROMNODE,TONODE) is negative.**

The junction modelling module has received a negative load value for a turn. This is not allowed. Such a situation can occur when you perform advanced multi-user-class assignments where multiple loads are used and manipulated in jobs.

**Invalid turn. To node of turn (FROMNODE,TONODE) is not found in arms.**

Idem as for the from node, but now for the to node of a turn.

**Invalid use of arm attribute 'enabled'. This attribute may only be 'false' on junctions with more than four arms.**

Arms may only be disabled for junctions with five or more arms. If you use this option for four or three arm junction, this message will be posted. See § 2.1.5 and § 2.1.8 for more information.

**It is not possible to calculate this junction. The junction is not initialised or an error has occurred during initialisation.**

The calculate function of the junction modelling module is called before the initialisation or something went wrong during the initialisation. This message only occurs when something is wrong with the software of your assignment model. Re-install the official software. If this doesn't help, contact your distributor or Omnitrans International for further assistance.

**It is not possible to generate lane groups for this junction. The junction is not initialised or an error has occurred during initialisation.**

The lane group generation function of the junction modelling module is called before the initialisation or something went wrong during the initialisation. This message only occurs when something is wrong with the software of your assignment model or your general OmniTRANS version. Re-install the official software. If this doesn't help, contact your distributor or Omnitrans International for further assistance.

**Maximum/Minimum cycle time is zero or negative.**

The maximum or minimum cycle time of a junction is not allowed to be zero or negative. Although the Junction Editor does not allow you to enter a zero or negative value for the maximum or minimum cycle time, it might be possible that the database contains a negative value, e.g. due to import or manual editing.

**Maximum/Minimum green time of turn (FROMNODE,TONODE) is zero or negative.**

The maximum or minimum green time of a turn is not allowed to be zero or negative. Although the Junction Editor does not allow you to enter a zero or negative value for the maximum or minimum green time, it might be possible that the database contains a negative value, e.g. due to import or manual editing.

**Minimum cycle time is greater than maximum cycle time.**

The minimum cycle time of a junction is not allowed to be greater than the maximum cycle time.

**Minimum green time is greater than maximum green time for turn (FROMNODE,TONODE).**

The minimum green time of a turn is not allowed to be greater than the maximum green time.

**Missing ... lane at arm from node FROMNODE.**

It is not allowed to leave out a lane to an arm that is not an one-way (entry) lane or does not have a banned turn for the regarding movement. The message reports for which movement a lane is missing:

- Left-turn;
- Right-turn;
- Straight.

You can either add the missing lane or add a turn prohibition for the regarding movement. See § 2.1.5 for more information.

**Missing turn. Turn (FROMNODE,TONODE) not found.**

The junction modelling module is missing a turn object. This message can occur when links attached to a junction are skipped during the assignment or when there is an inconsistency in the project database. If the latter is the case, perform a 'Check Database...' from the main Project menu.

**Redundant ... lane at arm from node FROMNODE.**

It is not allowed to define a lane to a one-way entry lane or for a movement with a turn prohibition on it. The message reports for which movement the redundant lane is configured:

- Left-turn;
- Right-turn;
- Straight.

You can either add the missing lane or add a turn prohibition for the regarding movement. See § 2.1.5 for more information.

## 4 Output

This Chapter gives detailed information about the output data that is generated by the junction modelling module and is stored in the project database by either OtTraffic or OtMadam<sup>7</sup>.

This document only describes the available data fields and the possibilities for visualisation and/or reporting. It does not explain how you can prepare all these visualisations and reports, since that is part of the general OmniTRANS manual. Only in very specific cases, e.g. to create two-coloured pie charts with the junction's v/c-ratio, in depth explanations are provided.

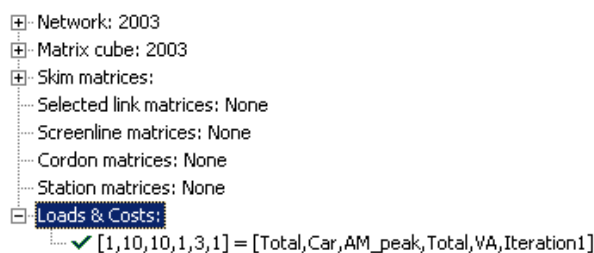


Figure 41 - Loads & Costs include junction output data.

Junction output data is stored as part of a 'load', for a particular pmturi.

The Chapter starts with an explanation of the data that is stored for the whole junction, i.e. is connected to the node object. Subsequently, the data that is connected to turns is discussed. The third paragraph explains the signal output data. The fourth paragraph gives a brief explanation about the lane data, which is not stored in the database directly, but is basically used as input for OtMadam's internal calculations to determine the link data. The Chapter ends with an overview of the database tables and fields related to the junction modelling output.

### 4.1 Junction Data

For each correctly calculated junction, the following general junction performance measures are calculated and stored in the project database:

- Critical v/c ratio;
- Weighed average v/c ratio;
- Average delay;
- Average queue;
- Level of service;
- Calculated cycle time.

These values are stored with the node object for a particular pmturi and can best be displayed by way of pie charts in the network or best be reported as a node based report.

---

<sup>7</sup> In OmniTRANS 4.2 OtMadam does not yet store the junction modelling output data in the project database. Junctions can best be analysed dynamically by looking at the load, dynspeed and density values at the link approaching the junction. See the OmniTRANS manual for more information.

### Pie Charts

Pie charts are displayed in the network and can be used to display all kind of node related data. Use the Pie Chart designer to display either junction input or output data that is related to the whole junction.

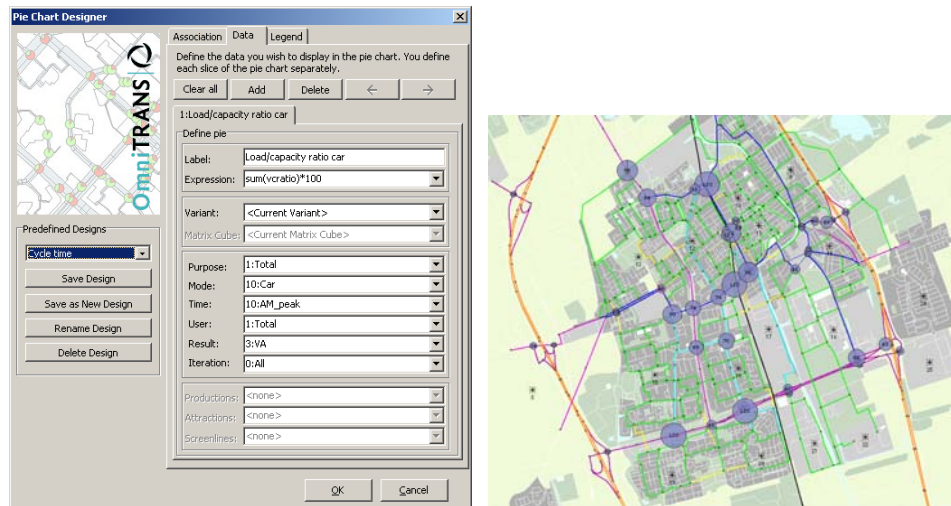


Figure 42 - Junction data in the Pie Chart Designer.

### Node Report

The junction data can also be reported as a report based on node data by way of using the Report Designer. The report is grouped by nodenr. Limit the report by connecting it to a selection which only holds the nodes that have been defined as a junction. The resulting report can then look like the one in Figure 44.

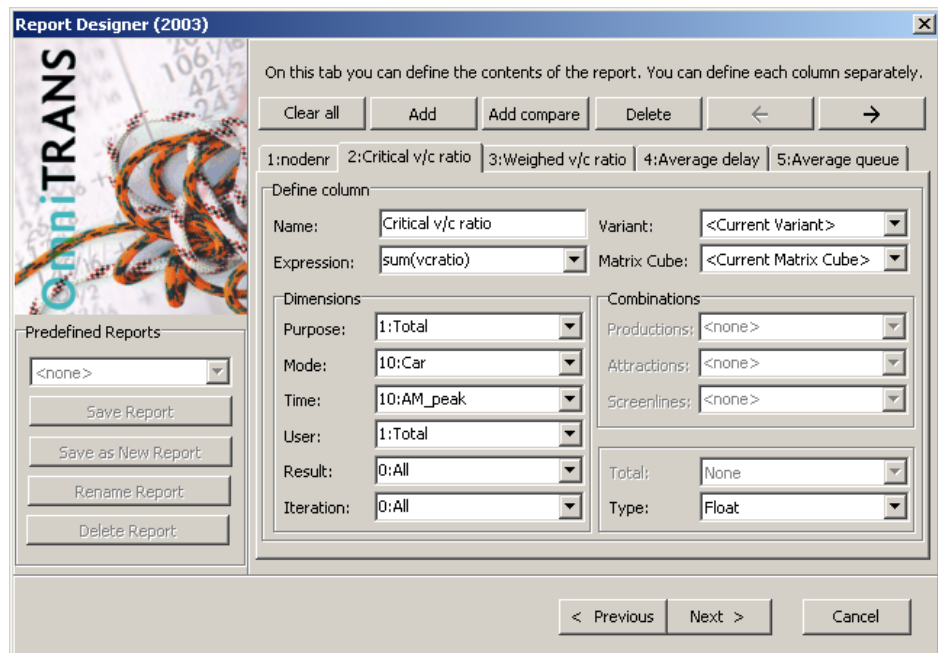


Figure 43 - Junction performance report in the Report Designer.

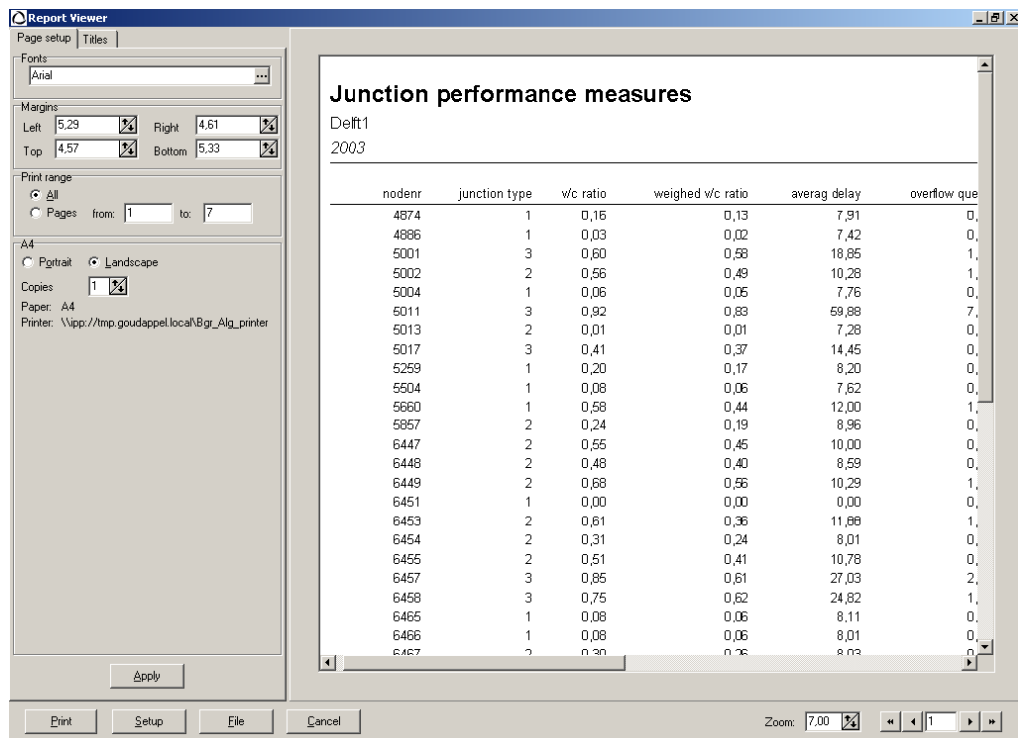


Figure 44 - Result of a junction performance measures report.

#### 4.1.1 Critical v/c ratio

For each turn on a junction a volume/capacity ratio is calculated. The highest value is stored as the critical v/c ratio (vcratio). In OmniTRANS 4.1 or earlier, this was the only junction performance measure available.

A high critical v/c ratio value gives a fast indication that at least one of the turning movements on the junction faces problems. Note that especially on yield or stop sign controlled junction, when the traffic flows for the major road are very high, the capacity of the turning movements which have to give easily decrease to the minimum 100 pcu's/hour. If the flow on these turning movements is 100 pcu's/hour or higher, the v/c ration exceeds the value of 1.0. Although most of the traffic of the junction doesn't face any problems, the v/c ratio of the junction is then above 1.0. For a more average measure, use the weighed v/c ratio.

A classical way of presenting the v/c ratio for junctions in OmniTRANS is by displaying it by way of a two coloured pie chart. The whole pie represents a value of 1.00 (or 100). The critical part is displayed red, while the remaining part is displayed green. Figure 45 shows an example of such a definition.

In the Pie Chart Designer, two parts are defined for the pie chart. The first part represents the normal v/c ratio, i.e. 'sum(vcratio)'. The second part is nothing more then '1-sum(vcratio)'. The first part is displayed with a red colour, the second with a green colour. All pies have an equal size.

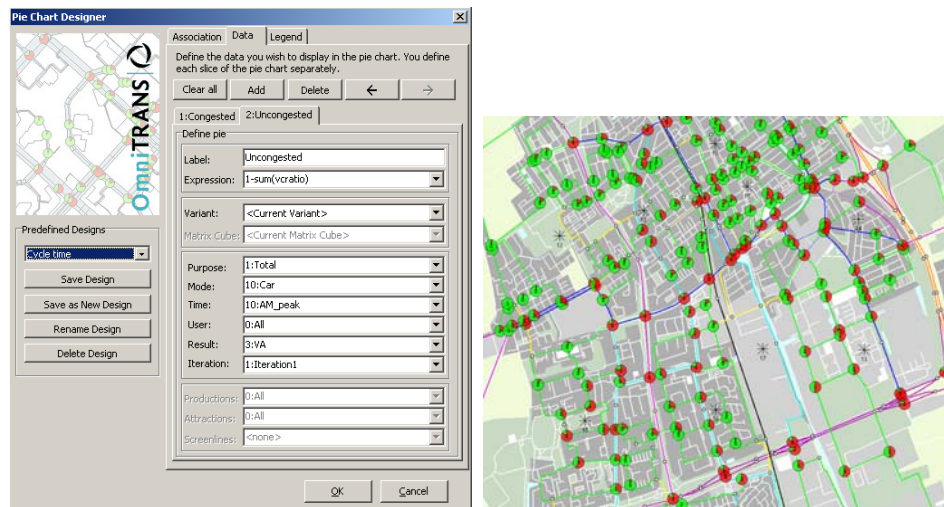


Figure 45 - Example of two-coloured pie chart presentation for v/c ratio's.

#### 4.1.2 Weighed average v/c ratio

For each turn on a junction a volume/capacity ratio is calculated. Subsequently, for each turning movement this value is multiplied with the volume (load) on that turn. These values are summed for the whole junction and divided by the sum of the loads. This produced a so called weighed average v/c ratio for the junction and is stored in the database in the field named 'wvcratio'.

The weighed average v/c ratio represents the average v/c ratio on the junction. Although the weighed value might look reasonable (e.g. lower than 0.8), this does not mean that there are no problems at the junction. Use the critical v/c ratio to see if there are turning movements with problems.

A classical way of presenting the v/c ratio for junctions in OmniTRANS is by displaying it by way of a two coloured pie chart. See the explanation of the critical v/c ratio in § 4.1.2.

#### 4.1.3 Average delay

For each turn on a junction a delay (seconds) is calculated. Subsequently, for each turning movement this value is multiplied with the volume (load) on that turn. These values are summed for the whole junction and divided by the sum of the loads. This produced a so called weighed average delay for the junction and is stored in the database in the field named 'delay'. For more information about the meaning of the delay value, see § 4.2.3.

#### 4.1.4 Average queue

For each turn on a junction a overflow queue length is calculated. Subsequently, for each turning movement this value is multiplied with the volume (load) on that turn. These values are summed for the whole junction and divided by the sum of the loads. This produced a so called weighed average overflow queue length for the junction and is stored in the database in the field named 'backofqueue'. For more information about the meaning of the overflow queue value, see § 4.2.4.

#### 4.1.5 Level of service

The general level of service value is not generated by the junction modelling module. This field can be used by the user to fill it manually or by way of a job.

#### 4.1.6 Calculated cycle time

For each signalised junction, a calculated cycle time is generated ('calccycletime'). If the signal type is automated, the cycle time is actually calculated by the junction modelling module. If the signal type is manually coded (or actuated), this value is simply copied from the input to the output tables of OmniTRANS.

It must be clear that the calculated cycle time ('calccycletime') can be something different than the input cycle time ('cycletime'). Both can be displayed as pie charts and used in node reports, but one is input while the other is output data.

The calculated cycle time can also be viewed on the Signal Settings tab of the Junction Editor. This will be explained in § 4.3.

### 4.2 Turn Data

For each correctly calculated junction, the following turning movement data is calculated and stored in the project database:

- Load (volume);
- Capacity;
- Cost (delay);
- Queue;
- Calculated green time;
- Calculated green time off set.

These values are stored with turn objects for a particular pmturi and can best be displayed by way of the Loads tab in the Junction Editor or best be reported as a turn based report.

#### Junction Editor

The Loads tab of the Junction Editor can be used to display the turning data on a junction. You can choose a particular value and the pmturi for which you want to see it, as well as a series of display settings.

The Junction Editor also has a separate Printing window in which you can control the layout, borders, titles and legend for printing.

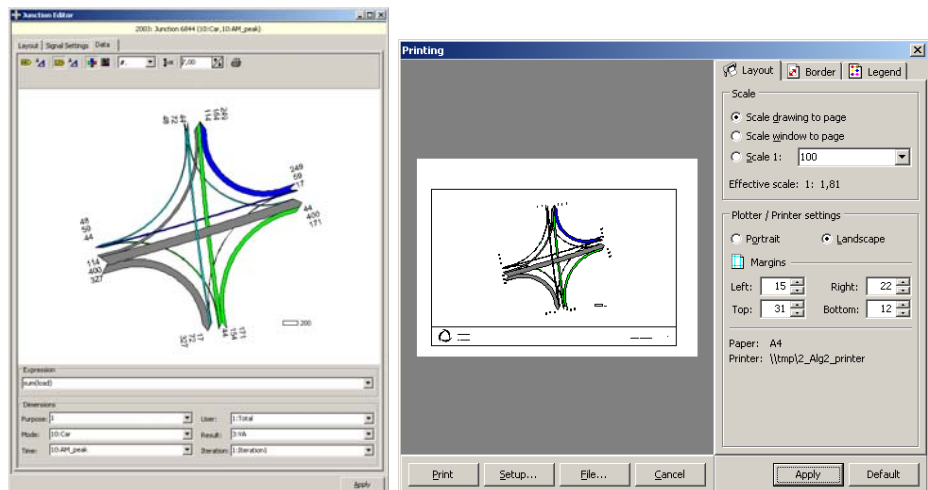


Figure 46 - Turning movement data in the Junction Editor.

### Turn Report

The turning movement data can also be reported as a report based on turn data by way of the Report Designer. The report should then be grouped by 'turnnr'. Limit the report by connecting it to a selection which only holds the turns for one particular junction (select all turn where 'pointnr' is equal to the 'noden' of the junction).

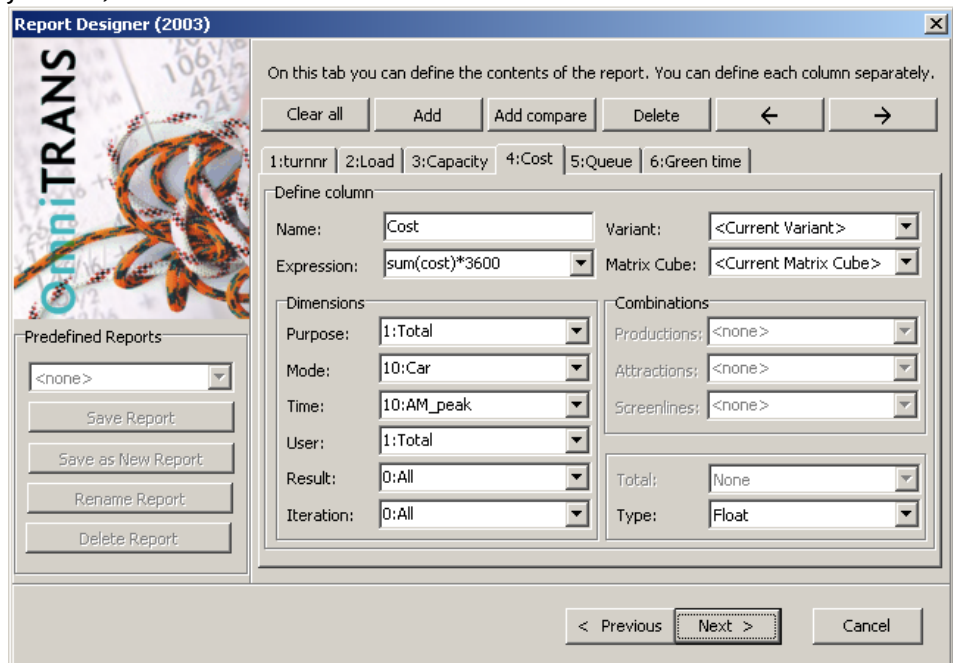


Figure 47 - Turning movements report in the Report Designer.

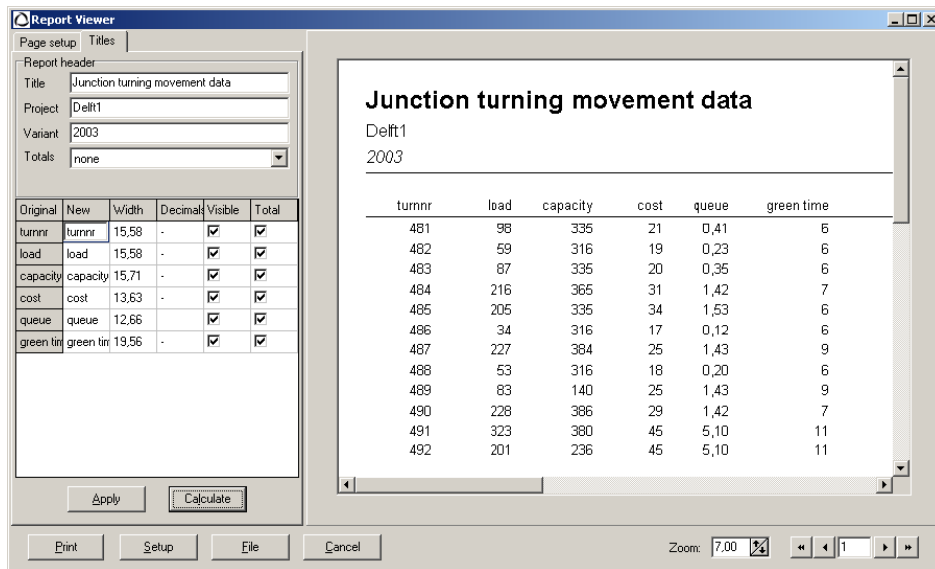


Figure 48 - Result of a junction turning movement report.

#### 4.2.1 Load

The load represents the volume of the flow for a particular turning movement. The units depend upon the chosen units for the OD-matrix that was assigned. If the OD-matrix contained vehicles for a two hour period, the units for the loads will be vehicles/two hour. Internally, the junction modelling module uses pcu's/hour, but by way of the junctionParameters property of OtAssign or OtMadam, the given units are converted when they enter and leave the module.

When a junction is defined as such, and a static assignment is performed with the junctions property set to false, the turning loads are still generated and stored in the database.

#### 4.2.2 Capacity

The capacity represents the maximum sustainable flow rate for a particular turning movement. The capacity includes effects of the type of movement, give way to other flows, blockade probabilities, number of lanes, sharing lanes, signal settings (ratio of cycle time and green time), etcetera. The units depend upon the chosen units for the OD-matrix that was assigned. If the OD-matrix contained vehicles for a two hour period, the units for the capacities will be vehicles/two hour. Internally, the junction modelling module uses pcu's/hour, but by way of the junctionParameters property of OtAssign or OtMadam, the given units are converted when they enter and leave the module.

For all Junction Types but roundabouts, the turning capacities can be used to determine the volume/capacity ratio for each turn, simply by typing the expression 'sum(load)/sum(capacity)' in the Junction Editor expression text box. For roundabouts this equation is not valid. In this case, the flows should be summed and one of the turning movement capacities should be taken (they are all the same). This is however not possible in the Junction Editor.

#### 4.2.3 Cost

The cost value represents the calculated average delay for a turning movement. The units are hours. The delay formulation includes uniform, incremental and geometric delay. Delay values are often represented in seconds, thus the Junction Editor offers an standard 'sum(cost)\*3600' expression.

#### 4.2.4 Queue

The queue value represent the so called overflow queue. This is the average number of vehicles per cycle left over at the end of green periods at signals or at the end of acceptable gap (unblock) periods during gap-acceptance periods. The units are dependent upon the whether pcu's or motor vehicles or something else is used in the OD-matrix that is assigned to the network.

#### 4.2.5 Calculated Green Time

The green time value ('calcgreentime') represents the average green time for the turning movement. Green times are always stored in seconds. If the signal type is automated, the green time is actually calculated by the junction modelling module. If the signal type is manually coded (or actuated), this value is simply copied from the input to the output tables of OmniTRANS.

The calculated green times can also be viewed on the Signal Settings tab of the Junction Editor. This will be explained in § 4.3.

#### 4.2.6 Calculated Green Time Off Set

The calculation of this value is not yet implemented and meant for future usage. The junction modelling module only determines the length of the green times, not the sequence en off sets for each signal group.

### 4.3 Signal Data

Cycle times can be displayed with pie charts in the network. Green times can be displayed on the Layout tab of the Junction Editor. Both types of data can also be displayed on the Signal Settings tab of the Junction Editor. This is the same tab where you can code the input data for your signal schemes.

Besides coding different signal schemes for different time period, you can also display the calculated values for a particular pmturi. You can choose a pmturi value from the Signalling Schemes list.

If you select a particular pmturi, the calculated green times and cycle time will be displayed. The green times are displayed in a grey-green colour, which means that they can not be modified. You can however, use the definition to start a new signalling scheme. OmniTRANS then copies the visible data to the new scheme and you can start to modify it.

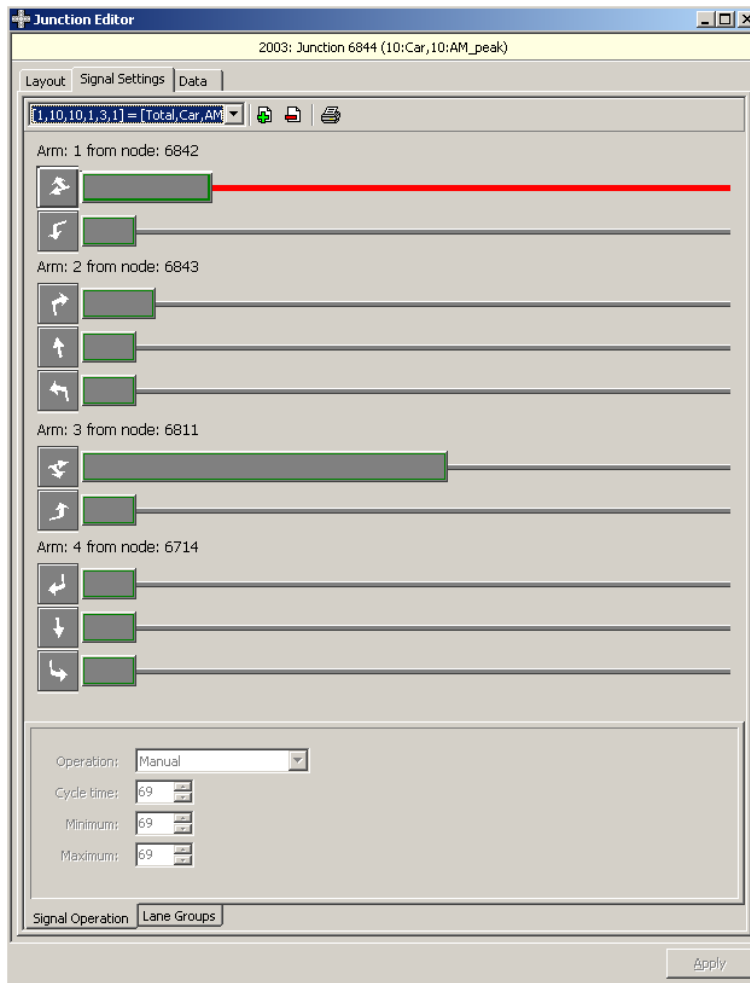


Figure 49 - Calculated signal data.

As you can see, the green times are all displayed with an offset value of zero. For the junction modelling module, this value is not (yet) important, and it is not calculated.

#### 4.4 Lane and Link Data for Dynamic Assignment

Without going into too much detail, it is good to mention that the junction modelling module also calculates a series of data for each approach lane separately. This is important data for the calculation of the situation on the segment approaching the junction for the dynamic assignment model OtMadam. This has been explained briefly in § 3.3.1.

For each lane, the following values are calculated:

- Load;
- Capacity;
- Delay;
- Queue;
- V/C ratio;
- Utilization over movements;

The first five values are similar to the ones that are calculated for each turning movement. The last one is extra. The junction modelling module performs a small equilibrium model which distributes the different turning flows over the available lanes, given the lane configuration (which movement can use which lane). This is done such that the delays are equal for each lane which is used by flows from one particular movement. This information is used by OtMadam to distribute the flows over the lanes.

The lane information is not stored in the project database. OtMadam aggregates the lane information to segment and then further to link information. For each link approaching a junction, the general values for flow (load), calculated speed (dyspspeed) and density (density) are used.

The situation on a junction can be displayed dynamically in the network, e.g. by showing the link bandwidths with the dyspspeed/freespeed ratio for the colour and the density for the width. See Figure 50 and Figure 51.

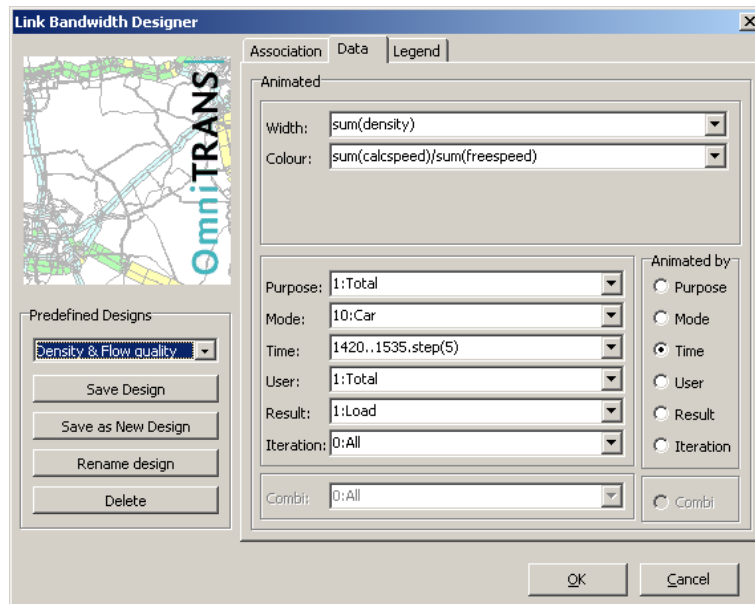


Figure 50 - Link bandwidth design with flow quality and density.



## 5 Methodology

This Chapter shows the formulas that are used for the calculation of the average delay and overflow queue length. Each junctiontype will be discussed seperately.

### 5.1 Symbols

$d$	Average control delay per vehicle (seconds)
$d_1$	Uniform delay (seconds)
$d_2$	Incremental delay (seconds)
$d_3$	Geometric delay (seconds)
$c$	Average cycle time (seconds)
$g$	Effective green time (seconds)
$Q$	Capacity (veh/h)
$s$	Saturation flow rate (veh/s)
$T$	Duration of analysis period (h)
$N_o$	Average overflow queue (veh)
$A$	Calibration factor for delay formulation, typically in the range 0.0-0.8
$a$	Calibration factor for queue formulation, typically in the range 0.0-0.8
$B$	Calibration factor for delay formulation, typically in the range 0.0-0.01
$b$	Calibration factor for queue formulation, typically in the range 0.0-0.01
$K$	Incremental factor that is dependent on controller settings for delay formulation, typically in the range 0.5-1.0
$k$	Incremental factor that is dependent on controller settings for queue formulation, typically in the range 0.5-1.0
$M$	Calibration factor for delay formulation, typically in the range 4-12
$m$	Calibration factor for queue formulation, typically in the range 4-12
$N$	Calibration factor for delay formulation, typically in the range -1-2
$n$	Calibration factor for queue formulation, typically in the range -1-2

### 5.2 Equal junction

The calculation of the capacity is done by lane group. In the case of an equal junction, each turn of each branch will be treated as a lane group. This means for a normal (4-way) junction, the number of lane groups is 12 (see Figure 52).

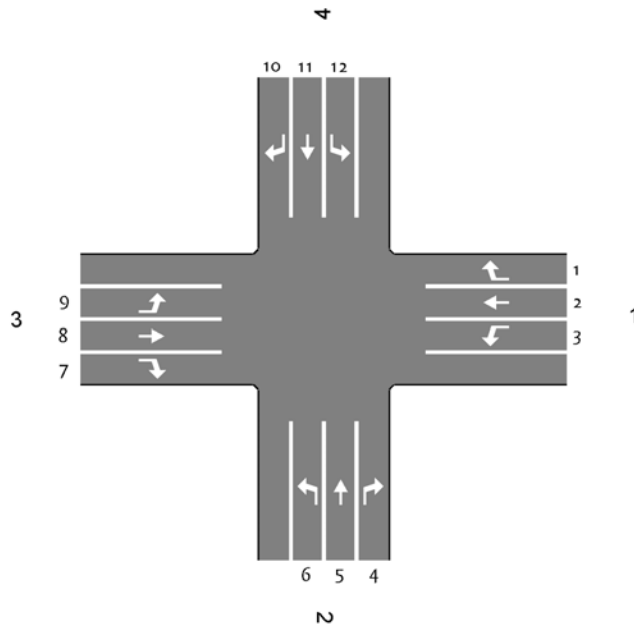


Figure 52: numbering of the lane groups for a 4-way equal junction.

### 5.2.1 Capacity

The calculation of the capacity is divided into two parts. The first part calculates the capacity per lane group. In the second part, the volume/capacity ratio of each lane group which has to give way is taken into account.

Part 1:

The capacity is calculated using the following formula (Bovy, 1991)

$$Q_{b,l_g} = \max \left( \frac{1}{\gamma} \delta (s_{l_g} - 0.99(\alpha l_{AC} + \beta l_C)), Q_{\min} \right)$$

$$l_{AC} = \sum_{l_g \in AC_{l_g}} l_{l_g} \tag{5.1}$$

$$l_C = \sum_{l_g \in C_{l_g}} l_{l_g}$$

With

- $Q_{b,lg}$  = base capacity of lane group  $lg$
- $Q_{\min}$  = minimal capacity
- $s_{lg}$  = saturation flow of lane group  $lg$
- $l_{AC}$  = load of the apparent-conflict movements
- $l_C$  = load of the conflict movements
- $l_{lg}$  = load of lane group  $lg$
- $AC_{lg}$  = set of lane groups that have an apparent-conflict with lane group  $lg$
- $C_{lg}$  = set of lane groups that have a conflict with lane group  $lg$

The values of the parameters  $\alpha, \beta, \gamma$  and  $\delta$  are calculated as follows:

$$\alpha = \alpha_b \left( 1 + \frac{1}{3} - \frac{2}{3} \sqrt{\frac{l_{AC}}{s_b} \frac{l_{AC}}{l_{AC} + l_C}} \right)$$

$$\alpha_b = 0.5$$

$$\beta = \begin{cases} 1 \rightarrow 1 \text{ exit lane on crossing road} \\ 0.7 \rightarrow 2 \text{ or more exit lanes on crossing road} \end{cases}$$

$$\gamma = \begin{cases} 1 \rightarrow 1 \text{ entry lane} \\ 0.6 \rightarrow 2 \text{ entry lanes} \\ 0.4 \rightarrow 3 \text{ or more entry lanes} \end{cases}$$

$$\delta = \begin{cases} \text{Right hand drive} = \begin{cases} 1 \rightarrow \text{right turn} \\ 0.8 \rightarrow \text{through movement} \\ 0.6 \rightarrow \text{left turn} \end{cases} \\ \text{Left hand drive} = \begin{cases} 1 \rightarrow \text{left turn} \\ 0.8 \rightarrow \text{through movement} \\ 0.6 \rightarrow \text{right turn} \end{cases} \end{cases} \quad (5.2)$$

The set of lane groups that have a conflict or an apparent-conflict are given by the means of a conflict matrix (for left-hand-drive-rule OmniTRANS uses another matrix). The way a conflict matrix is built is discussed in section 2.2.3.

In the calculation of  $\alpha$ , it is possible to have an apparent-conflict load of 0 and a conflict load of 0, this gives a division of zero by zero. In this case the limit goes to zero, so  $\alpha = \frac{2}{3}$ . In the determination of  $\beta$ , a value of 0.7 is used when the crossing road has more than one lane.

#### Part 2:

Now, based on the calculated capacities, the volume/capacity (v/c) ratio's can be calculated (only for the lane groups who has to give way are of any interest). If one of these ratio's exceeds 1.0 (hence a oversaturated situation), all capacities will be calculated again (part 1), where the loads of the lane groups with oversaturated situation will be adjusted according formula (5.3) (5% less load). This will be continued until no v/c ratio is oversaturated anymore or the maximum number of iterations has been reached (currently 100).

$$Q_{lg} = \begin{cases} Q_{b,lg} \rightarrow \frac{l_{lg}}{Q_{b,lg}} \leq 1 \\ Q_{b,new} \rightarrow \frac{l_{lg}}{Q_{b,lg}} > 1 \end{cases} \quad (5.3)$$

With

$Q_{lg}$  = capacity of lane group  $lg$

$Q_{b,lg}$  = base capacity of lane group  $lg$

$Q_{b,new}$  = new calculation of  $Q_{b,lg}$  with  $l_{lg,new}$

$$l_{lg,new} = \begin{cases} \frac{s_{lg}}{\gamma} \rightarrow l_{lg} > \frac{s_{lg}}{\gamma} \\ \frac{19}{20} l_{lg} \rightarrow Q_{b,lg} < l_{lg} \leq \frac{s_{lg}}{\gamma} \end{cases}$$

After this procedure, the lane group capacity will be used for calculate the lane capacity, using formula (5.4).

$$Q_l = \frac{1}{\sum_{i \in lg} \left[ \frac{l_i}{\sum_{j \in lg} l_j} \frac{1}{Q_i} \right]} \quad (5.4)$$

With:

$Q_l$  = Capacity per lane

$Q_i$  = Capacity on lane group  $i$

$l_i$  = Load on lane group  $i$

$l_j$  = Load on lane group  $j$

### 5.2.2 Delay

The calculation of the average delay uses two parameters for calculation, these are load and capacity. The calculation of the delay is given in formula (5.5).

$$D_l = \min(d_{1,l} + d_{2,l} + d_{3,l}, d_{\max,l}) \quad (5.5)$$

With

$D_l$  = Average delay on lane  $l$

$d_{1,l}$  = Uniform delay on lane  $l$

$d_{2,l}$  = Incremental delay on lane  $l$

$d_{3,l}$  = Geometric delay on lane  $l$

So the average control delay is the sum of the separate delays with a maximum of  $d_{\max,l}$ . The different  $d_{i,l}$  are all calculated in their own way and are given below.

The uniform delay is calculated in the way of:

$$d_{1,l} = \frac{3600}{Q_l} \quad (5.6)$$

With

$Q_l$  = Capacity on lane  $l$

The calculation of the capacity is treated in section 5.2.1. The incremental delay is calculated in the way of:

$$d_{2,l} = \begin{cases} 900T(Y_l)^N \left[ (Y_l - 1) + \sqrt{(Y_l - 1)^2 + \frac{MK(Y_l - Y_0)}{Q_l T}} \right] & \rightarrow Y_l > Y_0 \\ 0 & \rightarrow Y_l \leq Y_0 \end{cases} \quad (5.7)$$

With

$T$  = Duration of the project

$$Y_l = \frac{l_l}{Q_l}$$

$$Y_0 = a$$

The value of  $Y_0$  is 0.5, as will be seen later. This means that when the v/c ratio is smaller than 0.5, the value of 0 will be used. Or stated otherwise, there will be no incremental delay when the load is less than half the capacity. The default duration of the project is set on one hour, or 1.

The rest of the variables in this equation are defaults and their values are given in a table at the end of this chapter.

The calculation of the geometric delay is calculated as follows:

$$d_{3,l} = \begin{cases} 1 \rightarrow l_l > 0 \\ 0 \rightarrow l_l = 0 \end{cases} \quad (5.8)$$

### 5.3 Priority junction

The calculation of the capacity is done by lane group. In the case of an priority junction, each turn of each branch will be treated as a lane group. This means for a normal (4-way) junction, the number of lane groups is 12 (see Figure 53).

For calculation purposes, the junction will be rotated such that branch 1 (with lane groups 1,2 and 3) is always (on part of) the major road.

There are two basic situations for defining priority, straight- or turn-priority. Straight-priority means that the major road goes from branch 1 to branch 3 (and vice versa). For turn-priority, the major road goes from branch 1 to branch 2 (and vice versa).

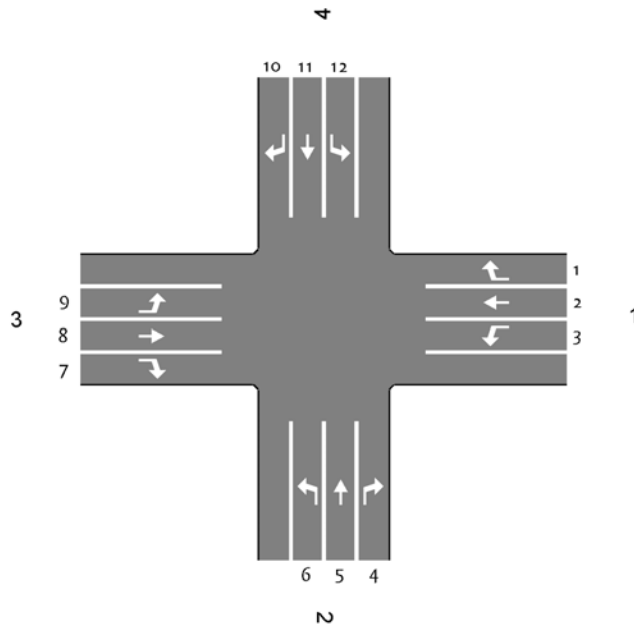


Figure 53: numbering of the lane groups for a 4-way priority junction.

### 5.3.1 Capacity

The calculation of the capacity is divided into two parts. The first part calculates the capacity per lane group. In the second part, the volume/capacity ratio of each lane group which has to give way is taken into account.

Part 1:

The capacity is calculated using the following formula (Bovy, 1991)

$$Q_{b,l_g} = \max \left( \frac{1}{\gamma} \delta (s_{l_g} - 0.99(\alpha l_{AC} + \beta l_C)) + c_{c1} + c_{c2}, Q_{\min} \right)$$

$$l_{AC} = \sum_{l_g \in AC_{l_g}} l_{l_g} \tag{5.9}$$

$$l_C = \sum_{l_g \in C_{l_g}} l_{l_g}$$

With

- $Q_{b,lg}$  = base capacity of lane group  $lg$
- $Q_{\min}$  = minimal capacity
- $s_{lg}$  = saturation flow of lane group  $lg$
- $l_{AC}$  = load of the apparent-conflict movements
- $l_C$  = load of the conflict movements
- $l_{lg}$  = load of lane-group  $lg$
- $AC_{lg}$  = set of lane groups that have an apparent-conflict with lane group  $lg$
- $C_{lg}$  = set of lane groups that have a conflict with lane group  $lg$

The values of the parameters  $\alpha, \beta, \gamma$  and  $\delta$  are calculated as follows:

$$\begin{aligned}
 \alpha &= \alpha_b \left( 1 + \frac{1}{3} - \frac{2}{3} \sqrt{\frac{l_{AC}}{s_b} \frac{l_{AC}}{l_{AC} + l_C}} \right) \\
 \alpha_b &= \begin{cases} 0 \rightarrow \text{seperate right turn lane on minor road} \\ 0.5 \rightarrow \text{no seperate right turn lane} \end{cases} \\
 \beta &= \begin{cases} 1 \rightarrow 1 \text{ exit lane on major road} \\ 0.7 \rightarrow 2 \text{ or more exit lanes on major road} \end{cases} \\
 \gamma &= \begin{cases} 1 \rightarrow 1 \text{ entry lane per lane group} \\ 0.6 \rightarrow 2 \text{ entry lanes per lane group} \\ 0.4 \rightarrow 3 \text{ or more entry lanes per lane group} \end{cases} \\
 \delta &= \begin{cases} \text{Rhd} = \begin{cases} 1 \rightarrow \text{major road} \\ 0.6 \rightarrow \text{through movement/left turn on minor road} \\ 0.8 \rightarrow \text{right turn on minor road} \end{cases} \\ \text{Lhd} = \begin{cases} 1 \rightarrow \text{major road} \\ 0.6 \rightarrow \text{through movement/right turn on minor road} \\ 0.8 \rightarrow \text{left turn on minor road} \end{cases} \end{cases} \quad (5.10)
 \end{aligned}$$

### **Influence of crossing multiple conflicting lanes ( $c_{c1}$ )**

The base capacity will be lowered with  $50/\gamma$ , who has to give way and has to cross at least two lanes of the concerning exit .

### **Influence of the width on the central reservation( $c_{c2}$ )**

When at least one vehicle can be placed on the crossing area, i.e. the width is larger than the length of one vehicle, the base capacity of several lane groups (who has to give way) will be heightened with  $100/\gamma$  .

The values of these capacity corrections ( $c_{c1}$  and  $c_{c2}$ ) are calculated as follows:

$$c_{c1} = \left\{ \begin{array}{l} \text{Rhd} \rightarrow \left\{ \begin{array}{l} \text{Straight priority} \rightarrow \left\{ \begin{array}{l} \lg 5,6 \text{ and exit lane} > 1 \text{ on branch 1} \rightarrow \frac{-50}{\gamma} \\ \lg 11,12 \text{ and exit lane} > 1 \text{ on branch 3} \rightarrow \frac{-50}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \\ \\ \text{Turn priority} \rightarrow \left\{ \begin{array}{l} \lg 8,12 \text{ and exit lane} > 1 \text{ on branch 2} \rightarrow \frac{-50}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \end{array} \right. \\ \\ \text{Lhd} \rightarrow \left\{ \begin{array}{l} \text{Straight priority} \rightarrow \left\{ \begin{array}{l} \lg 4,5 \text{ and exit lane} > 1 \text{ on branch 3} \rightarrow \frac{-50}{\gamma} \\ \lg 10,11 \text{ and exit lane} > 1 \text{ on branch 1} \rightarrow \frac{-50}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \\ \\ \text{Turn priority} \rightarrow \left\{ \begin{array}{l} \lg 7,11 \text{ and exit lane} > 1 \text{ on branch 1} \rightarrow \frac{-50}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \end{array} \right. \end{array} \right.$$
  

$$c_{c2} = \left\{ \begin{array}{l} \text{Rhd} \rightarrow \left\{ \begin{array}{l} \text{Straight priority} \rightarrow \left\{ \begin{array}{l} \lg 5,6 \text{ and } w_m \geq v_q \text{ on branch 1} \rightarrow \frac{100}{\gamma} \\ \lg 11,12 \text{ and } w_m \geq v_q \text{ on branch 3} \rightarrow \frac{100}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \\ \\ \text{Turn priority} \rightarrow \left\{ \begin{array}{l} \lg 8,12 \text{ and } w_m \geq v_q \text{ on branch 2} \rightarrow \frac{100}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \end{array} \right. \\ \\ \text{Lhd} \rightarrow \left\{ \begin{array}{l} \text{Straight priority} \rightarrow \left\{ \begin{array}{l} \lg 4,5 \text{ and } w_m \geq v_q \text{ on branch 3} \rightarrow \frac{100}{\gamma} \\ \lg 10,11 \text{ and } w_m \geq v_q \text{ on branch 1} \rightarrow \frac{100}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \\ \\ \text{Turn priority} \rightarrow \left\{ \begin{array}{l} \lg 7,11 \text{ and } w_m \geq v_q \text{ on branch 1} \rightarrow \frac{100}{\gamma} \\ \text{else} \rightarrow 0 \end{array} \right. \end{array} \right. \end{array} \right.$$

$w_m$  = Width of the median of the junction

$v_q$  = Space needed for a vehicle to queue

The set of lane groups that have a conflict or an apparent-conflict are given by the means of a conflict matrix (for left-hand-drive-rule OmniTRANS uses another matrix). The way a conflict matrix is built is discussed in section 2.2.3.

In the calculation of  $\alpha$ , it is possible to have an apparent-conflict load of 0 and a conflict load of 0, this gives a division of zero by zero. In this case the limit goes to zero, so  $\alpha = 0$  in case of a separate right turn lane and  $\alpha = \frac{2}{3}$  otherwise. In the determination of  $\beta$ , a value of 0.7 is used when the crossing road has more than one lane.

**Part 2:**

Now, based on the calculated capacities, the volume/capacity (v/c) ratio's can be calculated (only for the lane groups who has to give way are interesting). If one of these ratio's exceeds 1.0 (hence a oversaturated situation), all capacities will be calculated again (part 1), where all loads will be adjusted according formula (5.11) (5% less load). This will be continued until no v/c ratio is oversaturated anymore or the maximum number of iterations has been reached (currently 100).

$$Q_{lg} = \begin{cases} Q_{b,lg} \rightarrow \frac{l_{lg}}{Q_{b,lg}} \leq 1 \\ Q_{b,new} \rightarrow \frac{l_{lg}}{Q_{b,lg}} > 1 \end{cases} \quad (5.11)$$

With

$Q_{lg}$  = capacity of lane group  $lg$

$Q_{b,lg}$  = base capacity of lane group  $lg$

$Q_{b,new}$  = new calculation of  $Q_{b,lg}$  with  $l_{lg,new}$

$$l_{lg,new} = \begin{cases} \frac{s_{lg}}{\gamma} \rightarrow l_{lg} > \frac{s_{lg}}{\gamma} \\ \frac{19}{20} l_{lg} \rightarrow Q_{b,lg} < l_{lg} \leq \frac{s_{lg}}{\gamma} \end{cases}$$

After this procedure, the lane group capacity will be used for calculate the lane capacity, using formula (5.12).

$$Q_l = \frac{1}{\sum_{i \in lg} \left[ \frac{l_i}{\sum_{j \in lg} l_j} \frac{1}{Q_i} \right]} \quad (5.12)$$

With:

- $Q_l$  = Capacity per lane
- $Q_i$  = Capacity on lane group  $i$
- $l_i$  = Load on lane group  $i$
- $l_j$  = Load on lane group  $j$

### 5.3.2 Delay

The calculation of the average delay uses two parameters for calculation, these are load and capacity. The calculation of the delay is given in formula (5.13).

$$D_l = \min(d_{1,l} + d_{2,l} + d_{3,l}, d_{\max,l}) \quad (5.13)$$

With

- $D_l$  = Average delay on lane  $l$
- $d_{1,l}$  = Uniform delay on lane  $l$
- $d_{2,l}$  = Incremental delay on lane  $l$
- $d_{3,l}$  = Geometric delay on lane  $l$

So the average control delay is the sum of the separate delays with a maximum of  $d_{\max,l}$ . The different  $d_{i,l}$  are all calculated in their own way and are given below.

The uniform delay is calculated in the way of:

$$d_{1,l} = \frac{3600}{Q_l} \quad (5.14)$$

With

$$Q_l = \text{Capacity on lane } l$$

The calculation of the capacity is treated in section 5.3.1. The incremental delay is calculated in the way of:

$$d_{2,l} = \begin{cases} 900T(Y_l)^N \left[ (Y_l - 1) + \sqrt{(Y_l - 1)^2 + \frac{MK(Y_l - Y_0)}{Q_l T}} \right] & \rightarrow Y_l > Y_0 \\ 0 & \rightarrow Y_l \leq Y_0 \end{cases} \quad (5.15)$$

With

$$T = \text{Duration of the project}$$

$$Y_l = \frac{l_l}{Q_l}$$

$$Y_0 = a$$

The value of  $Y_0$  is 0.5, as will be seen later. This means that when the v/c ratio is smaller than 0.5, the value of 0 will be used. Or stated otherwise, there will be no incremental delay when the load is less than half the capacity. The default duration of the project is set on one hour, or 1.

The rest of the variables in this equation are defaults and their values are given in a table at the end of this chapter.

The calculation of the geometric delay is calculated as follows:

$$d_{3,l} = dp(1 - fth_l) + fth_l \quad (5.16)$$

With

$$fth_l = \frac{l_{th}}{l_l + l_{th} + l_{rt}}$$

$l_l$  = load on lane  $l$   
 $l_{lt}$  = load of left turn movements  
 $l_{th}$  = load of through movements  
 $l_{rt}$  = load of right turn movements

where  $fth_l$  is the fraction of through traffic. The parameter  $dp$  is a general parameter.

## 5.4 Signalized junction

The calculation of the capacity is done by lane group. In the case of an signalized junction, the software creates signal groups based on the layout of the junction. This means that for each branch one to three signal groups are defined, depending the definition of the lanes on this branch. For each signal group, the movements (left, straight or right) are determined. This means that the definition of lane groups is equal to signal groups in case of a signalized junction.

Slow traffic and bus lanes are taken into account when calculating the cycle time. However, in a pragmatic way.

### 5.4.1 Capacity

Each turn has a base capacity, which is equal to the saturation flow. Hence,

$$Q_{b,t} = s_t \quad (5.17)$$

With

$$\begin{aligned}
 Q_{b,t} &= \text{base capacity of turn } t \\
 s_t &= \text{saturation flow of turn } t
 \end{aligned}$$

Next, the base capacity per lane group can be calculated. This is done as follows:

$$\begin{aligned}
 Q_{b,l} &= \frac{\sum_{\text{turn } t \in l} Q_{b,t}}{nr_{t,l}} \\
 Q_{b,lg} &= \sum_{l \in \text{lanes}} Q_{b,l}
 \end{aligned} \quad (5.18)$$

With

$$\begin{aligned}
 Q_{b,l} &= \text{base capacity of lane } l \\
 Q_{b,lg} &= \text{base capacity of lane group } lg \\
 nr_{t,l} &= \text{number of turns } t \text{ on lane } l
 \end{aligned}$$

After the calculation of the base capacity, the loads are distributed over the lanes in such a way that every lane has about the same load-capacity ratio in the same signal groups. In mathematical notation:

$$\sum_{\text{turn } t \in \text{lane } l} \left( \frac{l_{t \in l}}{Q_{b,t}} \right) \approx \sum_{\text{turn } t \in \text{lane } j} \left( \frac{l_{t \in j}}{Q_{b,t}} \right), \forall l, j \quad (5.19)$$

With

$$l_{t \in l} = \text{load of turn } t \text{ on lane } l$$

The loads of the lane groups are simply the sum of the loads of the lanes.

Based on the conflict matrix, the normative maximum conflict group can be created.

The maximum conflict groups are essential for the planning of the green times of a traffic light. The groups consist of lane groups which are in conflict with all the other lane groups in the group. The creation of these maximum conflict groups happens in the following manner: per branch the lane groups are examined and compared with lane groups of other branches, when they are all in conflict, these lane groups together are a maximum conflict group. The normative maximum conflict group is the group with the highest signal cycle time. For the comparison of the cycle times, these are calculated in the following manner. First the base cycle-time is calculated in the way described in section 5.4.3. This calculation uses the variables  $t_{tl}$  and  $Y_t$  (the total loss time and the load/capacity ratio):

$$t_{tl} = t_{cl} + \sum_{p \in \text{phases}} (t_y - t_{end} + t_{start})$$

$$Y_t = \sum_{p \in \text{phases}} \left( \max_{l \in \text{lanes of } p} \left( \frac{l_l}{Q_{b,l}} \right) \right) \quad (5.20)$$

With

$Y_t$  = total load/capacity ratio

$t_{tl}$  = total loss time

$t_{cl}$  = total clearance time (sum of the inter-green times  
per lane group with extra for slow traffic and bus lanes)

After the computation of the maximum cycle-time, the green-times of the lane groups are calculated in the following way:

$$gt_p = \min \left( gt_{max}, \max \left( gt_{min}, \frac{Y_p}{Y_t} (Ct_{max} - t_{tl}) \right) \right) \quad (5.21)$$

With

$gt_p$  = green-time of phase  $p$

$Y_p$  = load/capacity ratio of phase  $p$

$Ct_{max}$  = base cycle-time

When one of the phases has either a shorter green-time than the minimum green-time or a longer green-time than the maximum green-time, the base cycle-time is calculated again with new variables; the variable  $t_{tl}$  is extended with the total minimum green-time and maximum green-time used and the variable  $Y_t$  is reduced

by the load/capacity ratio of those phases which have either a minimum green-time or a maximum green-time:

$$\begin{aligned} t_{tl,new} &= t_{tl} + t_{gt_{min}} + t_{gt_{max}} \\ Y_{t,new} &= Y_t - Y_{gt_{min}} - Y_{gt_{max}} \end{aligned} \quad (5.22)$$

With

$$\begin{aligned} t_{tl,new} &= \text{new total lost time} \\ t_{gt_{min}} &= \text{total time spent on minimum green-time} \\ t_{gt_{max}} &= \text{total time spent on maximum green-time} \\ Y_{t,new} &= \text{new load/capacity ratio} \\ Y_{gt_{min}} &= \text{total load/capacity of minimum green-time} \\ Y_{gt_{max}} &= \text{total load/capacity of maximum green-time} \end{aligned}$$

The cycle-times of the different maximum conflict groups can now be compared and the group with the highest cycle-time is the normative maximum conflict group. When multiple groups have the same maximal cycle-time, the group with the highest saturation degree is the normative maximum conflict group. The green-times have to be calculated again for those phases that have neither minimum nor maximum green-time.

$$gt_{p,new} = \frac{Y_p}{Y_{t,new}} (Ct_{new} - t_{tl,new}) \quad (5.23)$$

With

$$\begin{aligned} gt_{p,new} &= \text{adjusted green-time for 'normal' phases} \\ Ct_{new} &= \text{new cycle-time} \end{aligned}$$

The cycle-time and all green-times are known now and the saturation degree can be computed. This is done in the following way:

$$s_{deg,p} = Y_p \frac{Ct}{gt_{p,new}} \quad (5.24)$$

With

$$\begin{aligned} s_{deg,p} &= \text{saturation degree of phase } p \\ Ct &= \begin{cases} Ct_{max} \rightarrow \text{all green-times are between min and max} \\ Ct_{new} \rightarrow \text{some green-times are not between min and max} \end{cases} \end{aligned}$$

For all lane groups not in the normative conflict group, their green-times are calculated:

$$\begin{aligned} &\forall p \notin \text{normative conflict group:} \\ gt_p &= \min \left( gt_{max}, \max \left( gt_{min}, Y_p \frac{Ct}{s_{deg,max}} \right) \right) \end{aligned} \quad (5.25)$$

With

$$s_{deg,max} = \max_p (s_{deg,p})$$

The green-times of the lanes are the same as the green-times of the phases or lane groups they are on, so  $gt_l = gt_p, \forall l \in p$ . With the green-times of the lanes and the cycle-time known, the calculation of the capacity per lane is simply a weighted average:

$$Q_l = \frac{gt_{l \in p}}{Ct} Q_{b,l} \quad (5.26)$$

With

- $Q_l$  = capacity of lane  $l$
- $Q_{b,l}$  = base capacity of lane  $l$
- $gt_{l \in p}$  = green-time of lane  $l$  on phase  $p$
- $Ct$  = cycle-time

The saturation degree of the lanes is calculated by means of

$$s_{deg,l} = \frac{l_l}{Q_l} \quad (5.27)$$

With

- $s_{deg,l}$  = saturation degree of lane  $l$
- $l_l$  = load of lane  $l$

For turns, the saturation degree is

$$s_{deg,t} = \frac{\sum_{l \in \text{lanes}} l_{t \in l} \cdot s_{deg,l}}{\sum_{l \in \text{lanes}} l_{t \in l}} \quad (5.28)$$

With

- $s_{deg,t}$  = saturation degree of turn  $t$
- $l_{t \in l}$  = load of turn  $t$  on lane  $l$

The capacity of the turns can now be calculated as follows

$$Q_t = \frac{\sum_{l \in \text{lanes}} l_{t \in l}}{s_{deg,t}} \quad (5.29)$$

With

- $Q_t$  = capacity of turn  $t$

#### 5.4.2 Delay

The calculation of the average delay uses three parameters for calculation, these are load, capacity and green-time. The calculation of the delay for a signalized junction can be found in the signalized class and is the following

$$D = \min(d_1 + d_2 + d_3, d_{\max}) \quad (5.30)$$

With

- $D$  = Average delay
- $d_1$  = Uniform delay
- $d_2$  = Incremental delay
- $d_3$  = Geometric delay

So the average control delay is the sum of the separate delays with a maximum of  $d_{\max}$ . The different  $d_i$  are all calculated in their own way as given below and they are all calculated per lane.

The uniform delay is calculated in the way of:

$$d_1 = 0.5Ct \frac{\left(1 - \frac{gt_l}{Ct}\right)^2}{\left(1 - \left(\min\left(1, \frac{l_l}{Q_l}\right)\right) \frac{gt_l}{Ct}\right)} \quad (5.31)$$

With

- $Ct$  = cycle-time
- $gt_l$  = green-time of lane  $l$
- $l_l$  = load of lane  $l$
- $Q_l$  = capacity of lane  $l$

The calculation of the capacity, cycle-time and green-time is treated in section 5.4.1. The incremental delay is calculated in the way of:

$$d_{2,l} = \begin{cases} 900T \cdot Y_l^N \left[ (Y_l - 1) + \sqrt{(Y_l - 1)^2 + \frac{MK(Y_l - Y_0)}{Q_l T}} \right] & \rightarrow Y_l > Y_0 \\ 0 & \rightarrow Y_l \leq Y_0 \end{cases} \quad (5.32)$$

With

- $T$  = Duration of the project
- $Y_l = \frac{l_l}{Q_l}$
- $Y_0 = a + b \cdot s_{\text{sec},l} \cdot t_C$
- $s_{\text{sec},l} = \frac{Q_l}{3600}$

The value of  $x_0$  is 0.5, as will be seen later. This means that when the load divided by the capacity is smaller than 0.5, the value of 0 will be used. Or stated otherwise, there will be no incremental delay when the load is less than half the capacity. The duration of the project is normally one hour, or 1.

The rest of the variables in the equation are defaults and their values are given in a table at the end of this chapter.

The calculation of the geometric delay is calculated as follows:

$$d_{3,l} = dp(1 - fth_l) + fth_l \quad (5.33)$$

With

$$fth_l = \frac{l_{th}}{l_{lt} + l_{th} + l_{rt}}$$

$l_l$  = load on lane  $l$   
 $l_{lt}$  = load of left turn movements  
 $l_{th}$  = load of through movements  
 $l_{rt}$  = load of right turn movements

where  $fth_l$  is the fraction of through traffic. The parameter  $dp$  is a general parameter.

### 5.4.3 Calculation maximum cycle time

For the calculation of the maximum cycle time the load/capacity ratio of the deciding lane groups have to be known.

Two cycle times are calculated: then minimum and the optimal cycle time. The minimum cycle time is the cycle time that is minimal needed for taking care of all the traffic. The optimal cycle time, is the cycle time with minimum loss. However, sometimes the optimal is less than the minimum, which is not correct. Hence, the minimum will be used in that case.

The optimum cycle time is calculated by Webster's formula:

$$Ct_{opt} = \frac{F_1 \cdot t_{tl} + F_2}{1 - \frac{Y_t}{F_3}} \quad (5.34)$$

When the load/capacity ratio is less than the maximum allowed saturation degree a optimal cycle time is set equal to a given maximum cycle time. In mathematical notation this looks like:

$$Ct_{opt} = \begin{cases} \frac{F_1 \cdot t_{tl} + F_2}{1 - \frac{Y_t}{F_3}} \rightarrow Y_t < F_3 \\ Ct_{max,def1} \rightarrow Y_t \geq F_3 \end{cases} \quad (5.35)$$

$$t_{tl} = t_{cl} + \sum_{phases} (t_y - t_{end} + t_{start})$$

With

- $Ct_{opt}$  = optimal cycle time
- $Ct_{max,def1}$  = given maximum cycle time
- $Y_t$  = total load/capacity ratio
- $t_{tl}$  = total loss time
- $t_{cl}$  = total clearance time (sum of the inter-green times per lane-group with extra for slow traffic and bus lanes)

For slow traffic as well for busses an extra phase is created. This means that the total loss time will increase in both terms. Because, not only the number of phases

has increased, also the total clearance time has increased. The parameters not explained are constants and given at the end of this section.

The calculation for the minimum green time is done in the same way with the difference that the parameters are different. If it is higher, the minimum cycle time is calculated by Webster's formula. If it is lower the minimum cycle time is set equal to a given maximum cycle time. In formulation this is:

$$Ct_{min} = \begin{cases} \frac{t_{tl}}{Y_t} \rightarrow Y_t < s_{deg,max} \\ 1 - \frac{Y_t}{s_{deg,max}} \\ Ct_{max,def2} \rightarrow Y_t \geq s_{deg,max} \end{cases} \quad (5.36)$$

$$t_{tl} = t_{cl} + \sum_{phases} (t_y - t_{end} + t_{start})$$

With

$Ct_{min}$  = minimal cycle time

$Ct_{max,def2}$  = given maximum cycle time

$s_{deg,max}$  = practical saturation degree

$Y_t$  = total load/capacity ratio

$t_{tl}$  = total loss time

$t_{cl}$  = total clearance time (sum of the inter-green times per lane-group with extra for slow traffic and bus lanes)

The calculation of the maximum cycle time is now just the maximum of the values of the optimal cycle time and the minimal cycle time:

$$Ct_{max} = \min \left\{ Ct_{max,def3}, \max \{ Ct_{opt}, Ct_{min} \} \right\} \quad (5.37)$$

With

$Ct_{max}$  = maximal cycle time

$Ct_{max,def3}$  = given maximum cycle time

$Ct_{opt}$  = optimal cycle time

$Ct_{min}$  = minimal cycle time

The values of the given parameter are the following:

Variable	Description	Value
$F_1$	constant	1
$F_2$	constant	0
$F_3$	constant	0.85
$s_{deg,max}$	constant	0.85
$t_y$	yellow time	3
$t_{i-g}$	inter-green time	4
$t_{end}$	end-gain time	2
$t_{start}$	start-loss time	2
$Ct_{max,def 1}$	constant	120
$Ct_{max,def 2}$	constant	120
$Ct_{max,def 3}$	constant	120

(5.38)

### 5.5 Unsignalized roundabout

The calculation of the capacity is done by lane group. In the case of an unsignalized roundabout, each branch will be treated as a lane group. This means for a normal (4-way) unsignalized roundabout, the number of lane groups is 4 (see Figure 54).

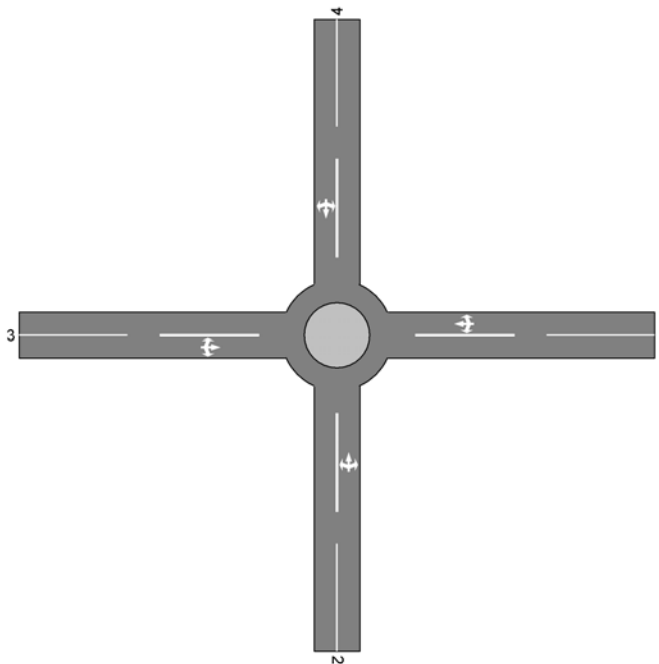


Figure 54: numbering of the lane groups for a 4-way unsignalized roundabout.

#### 5.5.1 Capacity

The calculation of the capacity is divided into two parts. The first part calculates the capacity per lane group.

In the second part, the volume/capacity ratio of each lane group which has to give way is taken into account.

The calculation of the capacity uses the loads of the lane groups. Because now the lane group is the complete branch, the loads of all turns of a branch has to be summed. These loads are computed by means of:

$$l_{lg} = \sum_{\text{turn } t \in \text{lane-group } lg} l_t \quad (5.39)$$

With

$l_{lg}$  = load of lane group  $lg$

$l_t$  = load of turn  $t$

Part 1:

The capacity is calculated using the following formula (Bovy, 1991).

$$Q_{b,lg} = \max \left( \frac{1}{\gamma} (s_{lg} - 0.99(\alpha l_{AC} + \beta l_C)), Q_{\min} \right)$$

$$s_{lg} = \frac{\sum_{\text{turn } t \in lg} s_{b,t}}{nr_{t,lg}} \quad (5.40)$$

$$l_{AC} = \sum_{lg \in AC_{l,lg}} l_{lg}$$

$$l_C = \sum_{lg \in C_{l,lg}} l_{lg}$$

With

$Q_{b,lg}$  = base capacity of lane group  $lg$

$Q_{\min}$  = minimum capacity

$s_{lg}$  = saturation flow of lane group  $lg$

$s_{b,t}$  = base saturation flow of turn  $t$

$nr_{t,lg}$  = number of turns  $t$  on lane group  $lg$

$l_{AC}$  = load of the apparent-conflict movements

$l_C$  = load of the conflict movements

$AC_{l,lg}$  = set of lanes that have an apparent-conflict with lane group  $lg$

$C_{l,lg}$  = set of lanes that have a conflict with lane group  $lg$

The values of the parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are calculated as follows:

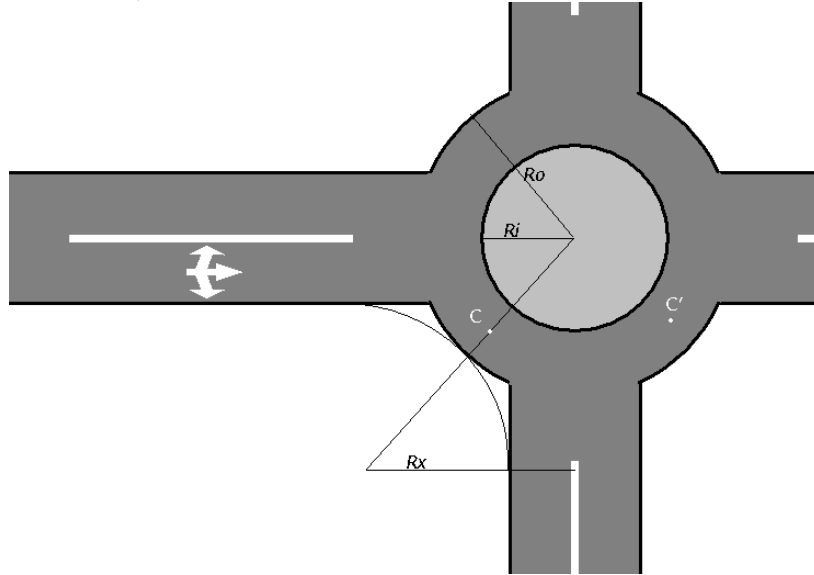
$$\alpha = \alpha_b \left( 1 + \frac{1}{3} - \frac{2}{3} \sqrt{\frac{l_{AC}}{s_b} \frac{l_{AC}}{l_{AC} + l_C}} \right)$$

$$\beta = \begin{cases} 1 \rightarrow 1 \text{ lane on roundabout} \\ 0.7 \rightarrow 2 \text{ or more lanes on roundabout} \end{cases} \quad (5.41)$$

$$\gamma = \begin{cases} 1 \rightarrow \text{else} \\ 0.65 \rightarrow 2 \text{ or more entry lanes and 2 or more lanes on roundabout} \end{cases}$$

Where especially  $\alpha_b$  some extra explanation requires.

Parameter  $\alpha_b$  depends on the distance between points C and C' over the roundabout, see picture below.



The calculation of the distance between point C and C' is the following:

$$d_{C,C'} = \begin{cases} R_{C,C'} (\angle_x + \angle_e) \rightarrow \text{if } \angle_x \text{ and } \angle_e \text{ exist} \\ 0 \rightarrow \text{else} \end{cases} \quad (5.42)$$

Where

$$R_{C,C'} = \frac{R_i + R_o}{2}$$

$$\angle_e = \arcsin \left( \frac{R_e + nr_{l,e} \cdot w_{l,e} + \frac{w_m}{2}}{R_e + R_o} \right)$$

$$\angle_x = \arcsin \left( \frac{R_x + nr_{l,x} \cdot w_{l,x} + \frac{w_m}{2}}{R_x + R_o} \right)$$

Where

$$R_o = R_i + nr_{l,ra} \cdot w_{l,ra}$$

With

$d_{C,C'}$  = distance over the roundabout between points  $C$  and  $C'$

$R_i$  = radius of circle inside the lanes of the roundabout

$R_o$  = radius of circle with the lanes of the roundabout

$R_e$  = radius of the entry circle

$R_x$  = radius of the exit circle

$nr_{l,e}$  = number of entry lanes

$nr_{l,x}$  = number of exit lanes

$nr_{l,ra}$  = number of lanes on roundabout

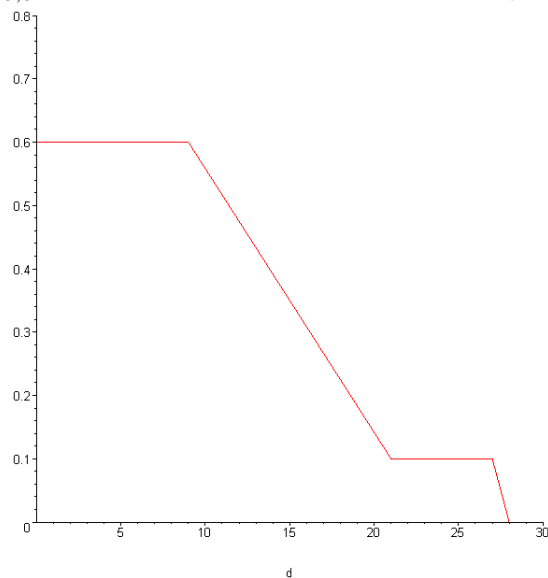
$w_{l,e}$  = width of entry lanes

$w_{l,x}$  = width of exit lanes

$w_{l,ra}$  = width of lanes on roundabout

$w_m$  = width of median

Part of the circle with radius  $R_x$  is drawn in the picture above. The circle with radius  $R_e$  lies on the other side of the branch and it has an other value. The way  $\alpha_b$  depends on  $d_{C,C'}$  is depicted in the following graph with  $\alpha_b$  on the vertical axis:



In mathematical notation this becomes:

$$\alpha_b = \begin{cases} 0.6 \rightarrow 0 \leq d_{C,C'} < 9 \\ \frac{23.4 - d_{C,C'}}{24} \rightarrow 9 \leq d_{C,C'} < 21 \\ 0.1 \rightarrow 21 \leq d_{C,C'} < 27 \\ \frac{28 - d_{C,C'}}{10} \rightarrow 27 \leq d_{C,C'} < 28 \\ 0 \rightarrow 28 \leq d_{C,C'} \end{cases} \quad (5.43)$$

In the calculation of  $\alpha$ , it is possible to have an apparent-conflict load of 0 and a conflict load of 0, this gives a division of zero by zero. In this case the limit goes to zero, so  $\alpha = \frac{4}{3}\alpha_b$ .

Part 2:

Now, based on the calculated capacities, the volume/capacity (v/c) ratio's can be calculated (only for the lane groups who has to give way are interesting). If one of these ratio's exceeds 1.0 (hence a oversaturated situation), all capacities will be calculated again (part 1), where all loads will be adjusted according formula (5.44) (5% less load). This will be continued until no v/c ratio is oversaturated anymore or the maximum number of iterations has been reached (currently 100).

$$Q_{lg} = \begin{cases} Q_{b,lg} \rightarrow \frac{l_{lg}}{Q_{b,lg}} \leq 1 \\ Q_{b,new} \rightarrow \frac{l_{lg}}{Q_{b,lg}} > 1 \end{cases} \quad (5.44)$$

With

$Q_{lg}$  = capacity of lane group  $lg$

$Q_{b,lg}$  = base capacity of lane group  $lg$

$Q_{b,new}$  = new calculation of  $Q_{b,lg}$  with  $l_{t,new}$

$$l_{t,new} = \begin{cases} \frac{s_{lg}}{\gamma \cdot l_{lg}} \rightarrow l_{lg} > \frac{s_{lg}}{\gamma} \\ \frac{19}{20} l_t \rightarrow Q_{b,lg} < l_{lg} \leq \frac{s_{lg}}{\gamma} \end{cases}$$

The calculation of the average delay uses two parameters for calculation, these are load and capacity and are both needed on lane level. The converting of capacity from lane group to capacity of lane is as follows:

$$Q_l = \begin{cases} \frac{l_l}{l_{lg}} Q_{lg} \rightarrow l_l > 0 \\ \frac{Q_{lg}}{nr_{l,lg}} \rightarrow l_l = 0 \end{cases} \quad (5.45)$$

With

$Q_l$  = capacity of lane  $l$

$Q_{lg}$  = capacity of lane group  $lg$

$l_l$  = load of lane  $l$

$l_{lg}$  = load of lane group  $lg$

$nr_{l,lg}$  = number of lanes  $l$  on lane group  $lg$

### 5.5.2 Delay

The calculation of the average delay uses two parameters for calculation, these are load and capacity. The calculation of the delay is given in formula (5.46).

$$D_l = \min(d_{1,l} + d_{2,l} + d_{3,l}, d_{\max,l}) \quad (5.46)$$

With

$$\begin{aligned} D_l &= \text{Average delay on lane } l \\ d_{1,l} &= \text{Uniform delay on lane } l \\ d_{2,l} &= \text{Incremental delay on lane } l \\ d_{3,l} &= \text{Geometric delay on lane } l \\ d_{\max,l} &= \text{Maximum delay on lane } l \end{aligned}$$

So the average control delay is the sum of the separate delays with a maximum of  $d_{\max,l}$ . The different  $d_{i,l}$  are all calculated in their own way and are given below.

The uniform delay is calculated in the way of:

$$d_{1,l} = \frac{3600}{Q_l} \quad (5.47)$$

With

$$Q_l = \text{Capacity on lane } l$$

The incremental delay is calculated in the way of:

$$d_{2,l} = \begin{cases} 900T \cdot Y_l^N \left[ (Y_l - 1) + \sqrt{(Y_l - 1)^2 + \frac{MK(Y_l - Y_0)}{Q_l T}} \right] & \rightarrow Y_l > Y_0 \\ 0 & \rightarrow Y_l \leq Y_0 \end{cases} \quad (5.48)$$

With

$$\begin{aligned} T &= \text{Duration of the project} \\ Y_l &= \frac{l_l}{Q_l} \\ Y_0 &= a \end{aligned}$$

The value of  $x_0$  is 0.5, as will be seen later. This means that when the load divided by the capacity is smaller than 0.5, the value of 0 will be used. Or stated otherwise, there will be no incremental delay when the load is less than half the capacity. The duration of the project is normally one hour, or 1.

The rest of the variables in the equation are defaults and their values are given in a table at the end of this chapter.

The geometric delay is:

$$d_{3,l} = dp \quad (5.49)$$

The last step in the calculation of delay, the delay of the lane groups is calculated by means of:

$$D_{lg} = \begin{cases} \frac{\sum_{l \in \text{lanes}} (D_l \cdot l_l)}{l_{lg}} \rightarrow l_{lg} > 0 \\ \frac{\sum_{l \in \text{lanes}} D_l}{nr_{l,lg}} \rightarrow l_{lg} = 0 \end{cases} \quad (5.50)$$

With

$D_{lg}$  = average delay of lane group  $lg$

$nr_{l,lg}$  = number of lanes  $l$  on lane group  $lg$

## 6 Average Overflow Queue

$$N_o = 0.25 QT x^n \left[ (x-1) + \sqrt{(x-1)^2 + \frac{mk(x-x_o)}{QT}} \right]$$

For signalised junctions the  $x_o$  is calculated like:

$$x_o = A + Bsg$$

For unsignalised junctions the  $x_o$  is calculated like:

$$x_o = A$$

## 7 References

The following list of references acted as a base for the junction modelling methodology in OmniTRANS.

- Akçelik, R. (1981), Traffic Signals: Capacity and Timing Analysis, Research Report ARR 31, Australian Road Research Board
- Akçelik, R. (1988), The Highway Capacity Manual Delay Formula for Signalized Intersections, ITE Journal, volume 48, nr. 3
- Bovy, P.H. (1991), Zusammenfassung des schweizerischen Kreisellhandbuchs, Strabe und Verkehr, nr. 3
- Braun, S.M. en J.N. Ivan (1996), Estimating Intersection Approach Delay Using 1985 and 1994 Highway Capacity Manual Procedures, Transport Research Report, nr. 1555, Transportation Research Board
- Brilon, W. en M. Grobmann (1991), The new German Guideline for capacity of unsignalized intersections, Highway Capacity and Level of Service, Balkema, Rotterdam
- Brilon, W. (1995), Delays at Oversaturated Unsignalized Intersections Based on Reserve Capacities, Transportation Research Board, Transportation Research Record 1484, National Academy Press, Washington D.C.
- Brilon, W. (1995), Wartezeiten an überlasteten Knotenpunkten ohne Lichtsignalanlagen, Strabenverkehrstechnik, nr. 11, 531-538
- Catling, I. (1977), A time-dependent approach to junction delays, Traffic Engineering & Control, november, 520-526
- Crabtree, M.R., R.A. Vincent en S. Harrison (1996), TRANSYT/10 User Guide, TRL Application Guide 28
- Hall, M.D., D. van Vliet en L.G. Willumsen (1980), Saturn a simulation-assignment model for the evaluation of traffic management schemes, Traffic Engineering and Control, april
- Harders, J. (1968), Die Leistungsfähigkeit nicht signalregelter städtischer Verkehrsknoten, Strabenbau und Verkehrstechnik, 76
- Kimber, R.M. en E.M. Hollis (1978), Peak-period traffic delays at road junctions and other bottlenecks, Traffic Engineering & Control, october, 442-445
- Kimber, R.M. en E.M. Hollis (1979), Traffic queues and delays at road junctions, Transport and Road Research Laboratory, TRRL Laboratory Report 909
- Leeuw, A.M. de (1997), Capaciteit van rotondes: ontwikkeling van nieuwe rekenmodellen, afstudeerverslag, Technische Universiteit Delft, Delft
- Shawaly, E.A.A., R. Ashworth en C.J.D. Laurence (1988), A comparison of observed, estimated and simulated queue lengths and delays at oversaturated signalized junctions, Traffic Engineering & Control, december, 637-643

- Stichting C.R.O.W. (1997), Publikatie Rotondes, Stichting Centrum voor Regelgeving en Onderzoek in de Grond-, Water- en Wegenbouw en de Verkeerstechniek, Ede
- Studiecentrum Verkeerstechniek (1986), Evaluatie verkeerslichtenregeling, mededeling 31, Studiecentrum Verkeerstechniek, Driebergen-Rijsenburg
- Teply, S. (1991), Quality of service in the new Canadian signal capacity guide, Highway Capacity and Level of Service, Balkema, Rotterdam, 377-386
- Transportation Research Board (2000), Highway Capacity Manual 2000, National Research Council, USA
- Trapp, K.H. (1977), Untersuchung der Leistungsfähigkeit vom Knotenpunkten und der Baukosteneinsparung beim stufenweisen Ausbau, Forschung Strabenbau und Strabenverkehrstechnik, Heft 230, Bundesminister für Verkehr
- Troutbeck, R.J. (1984), Capacity and Delays at Roundabouts - A Literature Review, Australian Research Board, 14(4), 205-215
- Vliet, D. van (1982), Saturn a modern assignment model, Traffic Engineering and Control, december
- Wu, N. (1992), Wartezeiten an festzeitgesteuerten Lichtsignalanlagen unter zeitlich veränderlichen (instationären) Verkehrsbedingungen, Strabenverkehrstechnik, nr. 3, 147-153